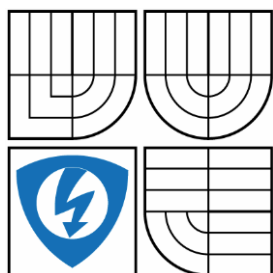


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF CONTROL AND INSTRUMENTATION

EVALUAČNÍ PLATFORMA PRO CAN TRANSCIEVERY

CAN TRANSCEIVER EVALUATION PLATFORM

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

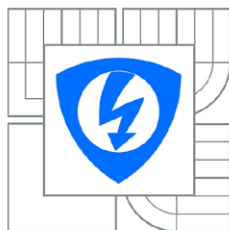
Bc. LADISLAV BAŽANT

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. PETR FIEDLER, Ph.D.

BRNO 2014



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav automatizace a měřicí techniky

Diplomová práce

magisterský navazující studijní obor
Kybernetika, automatizace a měření

Student: Bc. Ladislav Bažant
Ročník: 2

ID: 119354
Akademický rok: 2013/2014

NÁZEV TÉMATU:

Evaluační platforma pro CAN transceivery

POKYNY PRO VYPRACOVÁNÍ:

Cílem práce je navrhnout a vytvořit evaluační platformu pro testování CAN transceiverů, konkrétně:

1. Navrhněte prostorové uspořádání testovacích modulů, které budou simulovat typická zařízení v automobilu s ohledem na zamýšlený účel - testování funkčnosti nových obvodů.
2. Realizujte sadu testovacích modulů.
3. Vytvořte obslužný SW pro mikroprocesor, který bude realizovat komunikaci pro sběrnici CAN.
4. Vytvořte uživatelský SW pro PC, který umožní komfortní nastavení parametrů testované sítě.
5. Ověřte schopnost vytvořených modulů komunikovat na různých rychlostech a ověřte schopnost navázat a udržet komunikaci pro různá nastavení CAN transceiveru.

DOPORUČENÁ LITERATURA:

- [1] Communication & Systems Group, International Transceiver Conformance Test. CAN High-Speed Transceiver Conformance Test. Test Specification V1.0. 2004.
http://www2.csgroup.de/MunichGroupServer/public/documents/test_spec/HS_TRX_Test_Spec_V_1_0.pdf
- [2] CAN specification version 2.0, <http://esd.cs.ucr.edu/webres/can20.pdf>
- [3] FTZ e.V. an der Westsächsischen Hochschule Zwickau, Department of Electrical Engineering, EMC Investigations. EMC Evaluation of CAN Transceivers. Version 2.0 draft 1. 01-10-2004

Termín zadání: 10.2.2014

Termín odevzdání: 19.5.2014

Vedoucí práce: doc. Ing. Petr Fiedler, Ph.D.

Konzultanti diplomové práce:

doc. Ing. Václav Jirsík, CSc.
Předseda oborové rady

Abstrakt

Práce se zabývá návrhem testovacích modulů, které realizují komunikaci po CAN sběrnici. Tyto moduly je možné zapojit do sítě a následně testovat funkčnost nových CAN transceiverů. Součástí práce je obslužný software pro mikroprocesor a uživatelský software pro PC, pomocí kterého můžeme jednoduše měnit nastavení parametrů sítě a detekovat různé chyby při testování komunikace sítě.

Klíčová slova

Arbitrace, CAN, Controller Area Network, CAN budič, C8051F044, NCV7342 ON Semiconductor.

Abstract

The goal of this thesis is to design units with Control Area Network support. These modules can be connected into the network to test new CAN transceivers' functionality. Next part of the thesis is programming of a firmware for the microcontroller and a user program for PC which is used for network parameters' settings and for detecting various errors during communication.

Keywords

Arbitration, CAN, Controller Area Network, CAN Driver, C8051F044, NCV7342, ON Semiconductor.

Bibliografická citace:

BAŽANT, L. *Evaluační platforma pro CAN transceivery*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2013. 56s. Vedoucí diplomové práce byl doc. Ing. Petr Fiedler, Ph.D..

Prohlášení

„Prohlašuji, že jsem svou diplomovou práci na téma Evaluační platforma pro CAN transceivery vypracoval samostatně pod vedením vedoucího diplomové práce, s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne: **19. května 2014**

.....
podpis autora

Poděkování

Děkuji společnosti ON Semiconductor a především zaměstnancům Romanu Buzášovi, Ondřeji Kupčíkovi a Filipu Brtáňovi za jejich rady, trpělivost a materiální pomoc.

V Brně dne: **19. května 2014**

.....
podpis autora

Obsah

1	Úvod	9
1.1	Základní vlastnosti CAN protokolu	9
1.2	ISO normy	10
2	Fyzická vrstva	11
2.1	Časování a synchronizace	11
2.2	Délka vedení:	12
2.3	Terminace signálu	13
2.4	Použití ochranných obvodů	13
2.5	Výpočet časových parametrů	14
3	Linková vrstva	16
3.1	Přístup k médiu	16
3.2	Přenos zpráv	17
3.2.1	Datový rámec (Data Frame)	17
3.2.2	Žádost o data (Remote Data Frame)	18
3.2.3	Chybový rámec (Error Frame)	18
3.2.4	Zpráva o přetížení (Overload Frame)	19
3.3	Zabezpečení proti chybám	19
3.4	Detekce chyb	20
3.5	Programovací model C_CAN	21
3.5.1	CAN registry	22
3.6	Time-triggered CAN (TTCAN)	23
4	Koncepce	24
5	Návrh hardware	25
5.1	Mikroprocesor	25
5.2	Napájení	28
5.3	USB-UART Interface	29
5.4	CAN budič	30
5.5	CAN konektor	31
6	Návrh modulu	32
6.1	Propojení modulů	33
6.2	LED indikace	34
7	Software modulu	35
7.1	Software pro MCU	35

7.2	Ovládací software pro PC	38
7.3	Popis programu	38
7.4	Vnitřní struktura programu	39
7.5	Komunikace mezi PC a modulem.....	40
7.6	Podporované funkce.....	42
7.7	Nastavení rychlosti komunikace	45
8	Testování sítě.....	47
8.1	Testování komunikace	47
8.1.1	Měření číslo 1.....	47
8.1.2	Měření číslo 2.....	48
8.1.3	Měření číslo 3.....	48
8.1.4	Měření číslo 4.....	49
8.1.5	Měření číslo 5.....	50
8.1.6	Měření číslo 6.....	51
9	Závěr.....	52

1 ÚVOD

S nástupem digitálních řídicích jednotek pro osobní automobily na konci osmdesátých let vznikla potřeba vyvinout jednotný protokol pro diagnostiku těchto systémů. V počátcích využívali výrobci svoje vlastní komunikační protokoly a standardy, což bylo velkým problémem pro multi-značkové servisy, jelikož musely pořizovat velké množství specifických diagnostických přístrojů. Vývoj Control Area Network (CAN) protokolu začal v letech 1983 firmou Bosch. Protokol byl představen o 3 roky později a první CAN budiče byly uvedeny na trh v roce 1987 firmou Intel a Philips. V tuto dobu však ještě nebyla zveřejněna standardizace budičů. Situace se zlepšila začátkem devadesátých let, kdy se kladl důraz na normalizaci komunikačních protokolů. První CAN specifikace CAN 2.0A vznikla v roce 1991 a ISO norma 11898 byla uvolněna v roce 1993. Norma definuje fyzickou a datovou vrstvu sběrnice do rychlosti 1 Mbit/s. O dva roky později byla vytvořena nová specifikace CAN 2.0B, která se liší oproti původní verzi v délce identifikátoru zprávy. [1]

Sběrnice CAN se stala velice populární pro svoji jednoduchost, nízkou cenu a především pro dobré zabezpečení proti chybám. Ze začátku byla tato sběrnice používána pouze v automobilovém průmyslu, ale posléze se rozšířila i do ostatních odvětví, jako například do výtahů, inteligentních domů, průmyslových automatizací apod.

Z důvodu velké popularizace CAN sítě je dnes na trhu více než 11 výrobců s více než 200 CAN budiči. Tyto budiče nejsou identické, ale mohou vykazovat odlišnosti při kritických konfiguracích. Právě pro tyto případy je vhodné vytvořit univerzální testovací moduly, pomocí kterých bude možné nastavovat, testovat a měřit funkce a chování celé CAN sítě v závislosti na použití různých budičů.

1.1 Základní vlastnosti CAN protokolu

CAN je sériový, multimaster a multicast protokol. To znamená, že není předem pevně určena hlavní vysílací stanice. Uzel, jenž chce začít vysílat, podá zprávu do sítě a stane se po dobu vysílání masterem a po odeslání zprávy uvolní linku pro další uzly. Zpráva obsahuje identifikátor, z něhož ostatní uzly poznají, zda je tato zpráva určena pro ně, nebo jestli ji mají ignorovat. Tento identifikátor také určuje prioritu daného uzlu, pokud dva uzly začnou vysílat ve stejný okamžik. Každý modul může mít nastaveny filtry, které určují, jaké zprávy se mají aktivně zpracovávat a které naopak ignorovat. Pro použití v moderních dopravních prostředcích je nutná především vysoká spolehlivost systému, kterou zabezpečuje celkem pět kontrolních mechanismů uvedených níže. [2]

1.2 ISO normy

Tato práce se odkazuje na CAN normy. Pro lepší přehlednost jsou zde uvedeny normy, které vznikly v průběhu vývoje CAN sběrnice. Normy jsou každých 5 let revidovány a nabízeny k prodeji na internetových stránkách mezinárodní organizace pro standardizaci. [3]

ISO 11898-1:2003 – původní revidovaná norma z roku 1993, specifikuje linkovou vrstvu a popisuje fyzické vedení signálu Controller Area Network (CAN) sítě. Tento dokument popisuje obecnou architekturu CANu v závislosti na hierarchických vrstvách podle ISO modelu. Dále poskytuje detailní informace o podvrstvách LLC a MAC.

ISO 11898-2:2003 – nejpoužívanější fyzický standard pro CiA.[4] Tato norma stanovuje vysokorychlostní přenos (rychlost do 1 Mbit/s) medium access jednotky (MAU) a medium dependent rozhraní (MDI) v závislosti na normě ISO 8802-3, která zahrnuje fyzickou vrstvu CAN.

ISO 11898-3:2006 – specifikuje bezpečný přenos dat odolný proti poruchám, dále MDI rozhraní pro komunikaci jednotek v osobních automobilech v přenosové rychlosti od 40 do 125 kbit/s.

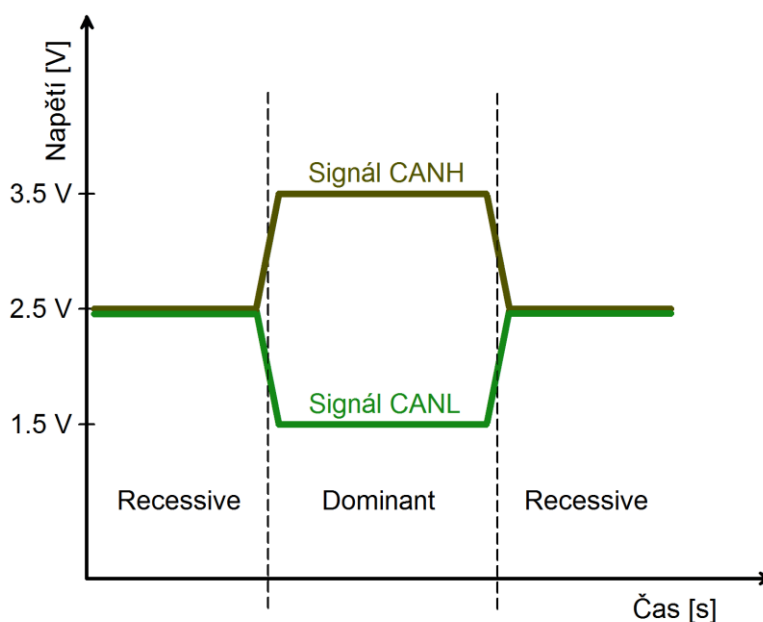
ISO 11898-4:2004 – stanovuje komunikaci na základě událostí (Time-Triggered). Norma je použitelná pro výměnu digitálních informací mezi elektronickými řídicími jednotkami v automobilech za použití časové synchronizace.

ISO 11898-5:2007 – určuje fyzickou vrstvu pro přenosovou rychlost do 1 Mbit/s pro použití v automobilech. Norma vychází z ISO 11898-2 a rozšiřuje požadavky pro úsporný mód, pokud není aktivní komunikace.

ISO 11898-6:2013 – určuje fyzickou vrstvu pro přenosovou rychlost do 1 Mbit/s pro použití v automobilech. Rozšiřuje normy ISO 11898-2 a ISO 11898-5 s využitím selektivního wake-up mechanismu v konfiguračním CAN rámci.

2 FYZICKÁ VRSTVA

Norma ISO-11898-2 CAN protokolu definuje dva stavy na sběrnici: **dominant** a **recessive**. Sběrnice zastává funkci logického součinu [5]. To znamená, že pokud všechny uzly vysílají stav recessive, tak i na sběrnici bude stav recessive. Naopak, pokud alespoň jeden uzel bude vysílat dominant, tak výsledný stav na sběrnici bude dominant. Fyzikální provedení není jasně definováno a může se lišit podle dané realizace. Pro realizaci přenosového média se využívá diferenciální sběrnice, kterou tvoří dva vodiče s označením **CANH** a **CANL**. Stav na sběrnici odpovídá rozdílu napětí V_{diff} na těchto dvou vodičích. Pro stav recessive je $V_{\text{diff}} = 0$ V, naopak pro stav dominant je $V_{\text{diff}} = 2$ V.



Obrázek 1: Stavy na sběrnici

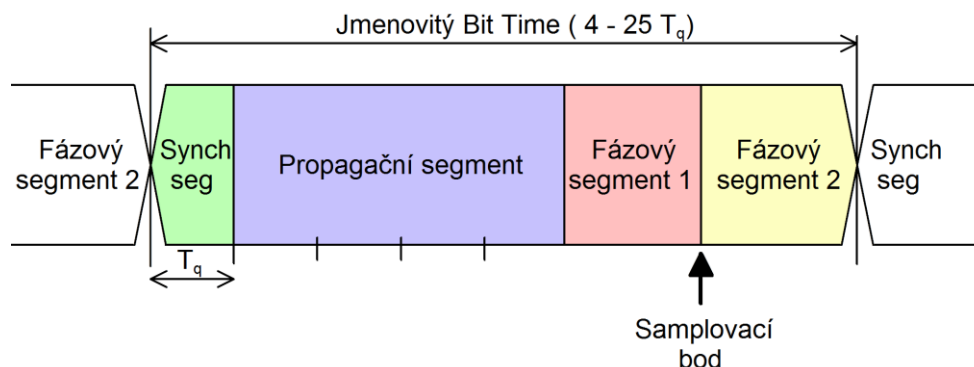
2.1 Časování a synchronizace

Každý bit je rozdělen do čtyř segmentů. Synchronizační, propagační a dva fázové segmenty. Každý segment se skládá z jednoho nebo více časových kvant. Časové kvantum t_q je nejmenší jednotka, která je odvozena z frekvence krystalu a CAN děličky. Suma všech časových kvant se rovná délce jednoho bitu (Bit Time). Bit Time může nabývat od 4 do 25 t_q .

Synchronizační segment slouží k synchronizaci různých modulů na sběrnici. Velikost tohoto segmentu se rovná jednomu t_q , počínaje začátkem odesílaného bitu na sběrnici.

Propagační segment kompenzuje zpoždění v důsledku dlouhého vedení sběrnice. Lze nastavit délku propagačního segmentu od 1 do 8 t_q .

Fázový segment 1 a Fázový segment 2 mohou být zkráceny nebo prodlouženy v závislosti na synchronizaci SJW. Každý segment může nabývat hodnot od 1 do 8 t_q . Mezi fázovými segmenty se nachází samplovací bod, ve kterém by měla být čtena ustálená hodnota na sběrnici.



Obrázek 2: Časování a synchronizace

2.2 Délka vedení:

Pro eliminaci odrazů na sběrnici se využívá dvou zakončovacích rezistorů o nominální hodnotě 120 Ω (od 95 Ω do 140 Ω). Norma doporučuje použití stíněných kabelů pro lepší EMC imunitu, ale pro normální použití to není nezbytné.

Pro maximální rychlost 1 Mbit/s norma udává maximálně 40 metrů dlouhé vedení. Pro delší vedení je nutné snížit přenosovou rychlost. Pokud uvažujeme o rychlosti šíření signálu 5 ns/m, tak pro 100 metrů dlouhé vedení je doba přenosu signálu od vysílače k přijímači a zpět rovna 1000 ns. Konzervativní pravidlo pro maximální povolenou přenosovou rychlost v závislosti na délce sběrnice se udává, jako součin rychlosti přenosu a délky sběrnice, který musí být menší nebo roven 50. [8]

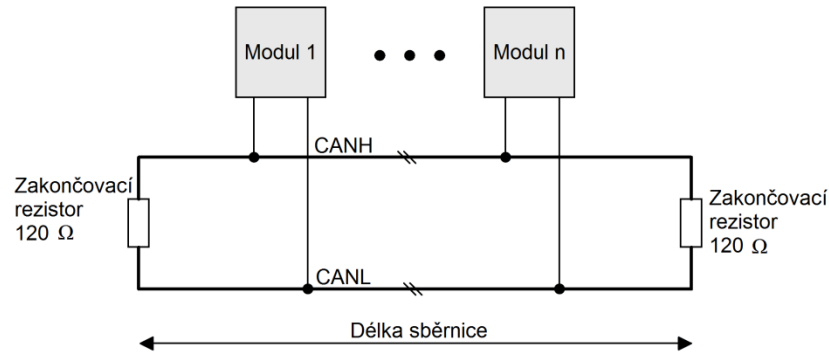
$$\text{Rychlost přenosu (Mbit/s)} \times \text{Délka sběrnice} \leq 50 \quad (1)$$

Délka vedení (m)	Maximální rychlost (Mbit/s)
40	1,00
100	0,50
200	0,25
500	0,10
1000	0,05

Tabulka 1: Doporučené komunikační rychlosti při určitých délkách vedení

2.3 Terminace signálu

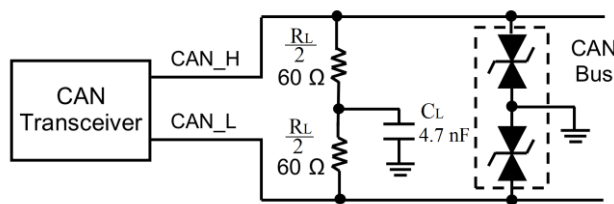
K potlačení odrazů na sběrnici při použití 120 Ω kabelu je nutné použít dvou zakončovací rezistorů o stejné nominální hodnotě. Rezistory by měly být připojeny na nejvzdálenějších místech sítě, pokud máme rozvětvenou síť.



Obrázek 3: Příklad jednoduché terminace

Lze také použít druhý způsob terminace, a to split terminaci. Split terminace, zobrazena na obrázku číslo 4, je vhodná pro stabilizaci napětí na sběrnici v recesivním stavu, a také jako dolní pásmová propust. Pokud mají dva rezistory stejnou hodnotu, toto zapojení je také účinné pro zvýšení elektromagnetické imunity a kompatibility. Při hodnotě kondenzátoru $C_L = 4.7 \text{ nF}$ je mezní frekvence f_C podle vztahu 2 rovna 565 kHz tzn., že filtr můžeme použít pro rychlosti menší než 1.13 Mbit/s.

$$f_C = \frac{1}{2\pi R C_L} \quad (2)$$



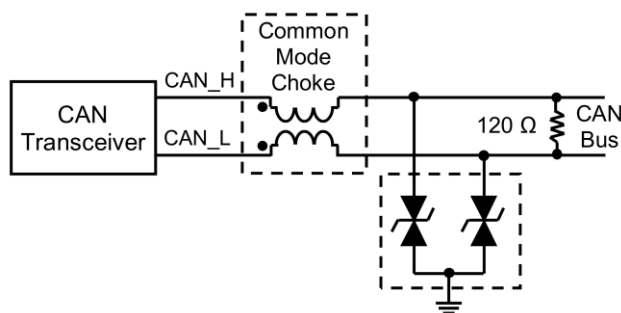
Obrázek 4: Split terminace [9]

2.4 Použití ochranných obvodů

Pro vytvoření robustního systému je vhodná kombinace několika prvků. Split terminace je popsána výše, dále je možné použít TVS diodu (Transil) nebo pro některé případy je doporučeno použít tlumivku souhlasného napětí (Common Mode Choke). Transil slouží k pohlcení energie při velké napětíové špičce. Tyto diody mají rychlou dobu náběhu, méně než 1 ns, takže napětí na sběrnici by nemělo překročit jmenovité napětí CAN budiče. TVS diody jsou podobné zenerovým diodám s tím rozdílem, že mají větší

PN přechod a jsou navrženy pro překlenutí napěťových špiček namísto stabilizace napětí. Schéma zapojení TVS diody je zobrazeno na obrázku číslo 4 a 5.

Použití cívky souhlasného napětí podle obrázku 5 zeslabuje superponovaný šum na obou stranách datové linky. Cívka poskytuje vysokou impedanci pro souhlasné signály a nízkou impedanci pro rozdílné signály. Cívky jsou efektivní nástroje pro implementování filtrování bez velkého zkreslení. Cívka může být kombinována s TVS diodou. [9]



Obrázek 5: Použití cívky na sběrnici [9]

2.5 Výpočet časových parametrů

Minimální čas propagačního zpoždění pro zajištění správné samplovací hodnoty se udává jako součet propagačního zpoždění z jednoho modulu na druhý a zpět, kde se moduly nachází A a B na sběrnici na nejvzdálenějším místě.

$$t_{PROP_SEG} = t_{PROP(A,B)} + t_{PROP(B,A)} \quad (3)$$

Propagační zpoždění z modulu A do modulu B je dáno jako součet zpoždění na sběrnici, zpoždění na vysílacím budiči A a zpoždění na přijímacím budiči B.

$$t_{PROP(A,B)} = t_{TX(A)} + t_{BUS(A,B)} + t_{RX(B)} \quad (4)$$

Úpravou těchto rovnic dostaneme:

$$t_{PROP_SEG} = 2(t_{BUS} + t_{TX} + t_{RX}) \quad (5)$$

Velikost propagačního segmentu se rovná:

$$PROP_SEG = ROUND_UP\left(\frac{t_{PROP_SEG}}{t_Q}\right) \quad (6)$$

Postup pro zjištění časových parametrů sběrnice: [16]

- 1) Zjištění minimální povolené doby pro propagační segment, který se získá ze zpoždění CAN budičů a zpoždění signálu na sběrnici;
- 2) vybrání časování pro CAN řadič. Podle dostupných zdrojů časování se zjistí velikost časového kvanta. V závislosti na zamýšlené rychlosti CAN sběrnice se podílem periody rychlosti sběrnice a délky časového kvanta zjistí počet časových kvant pro jeden bit;
- 3) spočítání velikosti propagačního segmentu z rovnice číslo 5. Pokud je výsledek větší než počet časových kvant je nutné zvětšit časování CAN sběrnice.

- 4) Určení velikosti fázových segmentů: Z celkového počtu časových kvant se odečte velikost propagačního segmentu a jedno časové kvantum t_Q pro synchronizační segment. Pokud je tento výsledek menší než 3, přejdeme zpět na krok 2 a zvolí se rychlejší časování sběrnice. Pokud je výsledek liché číslo a větší než 3, tak se jednička přičte k propagačnímu segmentu a zbytek se podělí dvěma a přiřadí se k oběma fázovým segmentům. Pokud je výsledek roven 3, tak se tři časová kvanta rozdělí v poměru 1:2 pro fázový segment 1 a 2. V ostatních případech se výsledek vydělí dvěma a přiřadí se k oběma fázovým segmentům.
- 5) Určení velikosti SJW. Hodnota SJW musí být v intervalu fázového segmentu 1 a maximální velikosti 4.
- 6) Vypočítání požadované přesnosti oscilátoru pomocí následujících rovnic, kde NBT je počet časových kvant.

$$\Delta f < \frac{t_{SJW}}{10 \times NBT} \quad (7)$$

$$\Delta f < \frac{\text{MIN}(\text{fáz. seg1, fáz. seg2})}{2(13 \times NBT - \text{fáz. seg2})} \quad (8)$$

Požadovaná přesnost je vždy menší hodnota z vypočtených výsledků.

3 LINKOVÁ VRSTVA

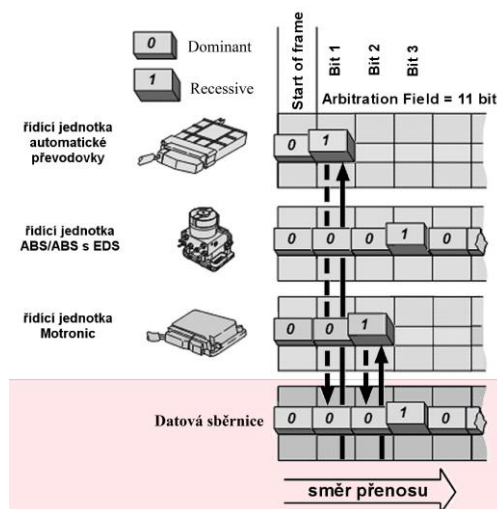
Linková vrstva se skládá z podvrstev MAC (Media Access Control) a LLC (Logical Link Control). **MAC** se stará o rámcování zpráv, arbitraci komunikace, detekci a signalizaci chyb, a acknowledge mechanismus. Je to hlavní část specifikace CAN protokolu. **LLC** je zodpovědná za přenos bitů ze zdroje do cíle, filtrování příchozích zpráv, datový přenos z remote data request.

3.1 Přístup k médiu

Pokud je volná sběrnice, může libovolný uzel zahájit vysílání. Pokud uzel zahájí vysílání dříve než ostatní uzly na sběrnici, získá sběrnici pro sebe a ostatní uzly musí čekat, až se ukončí vysílání. Jedinou výjimkou jsou chybové rámce, které může libovolný uzel vyslat, pokud detekuje chybu v přenášené zprávě.

V případě, že dva uzly začnou vysílat současně, získá přístup na sběrnici ten uzel, který má větší prioritu udávanou nižším identifikátorem. Identifikátor je uveden na začátku zprávy. Vysílač porovnává postupně odeslané bity s hodnotami na sběrnici, a pokud je na sběrnici dominantní hodnota a vysílač měl odeslat recesivní hodnotu, znamená to, že tento vysílač má menší prioritu a ihned přeruší další vysílání. Přístup na sběrnici získá vysílač s nejvyšší prioritou, který dokončí vysílání.

Na obrázku číslo 6 je naznačena situace současného vysílání tří jednotek s různou prioritou (identifikátorem). Řídicí jednotka automatické převodovky má nejvyšší identifikátor (nejnižší prioritu) a po prvním odeslaném bitu v identifikátoru přestane vysílat na sběrnici podle výše uvedeného přístupu. Po druhém bitu se zjistí, že řídicí jednotka ABS/ABS s EDS má menší prioritu než řídicí jednotka ABS, a tudíž také přestane vysílat. Arbitraci vyhrála řídicí jednotka ABS, která má nejmenší číslo identifikátoru (nejvyšší prioritu) a může dokončit odesílání celé zprávy.



Obrázek 6: Arbitrace odesílané zprávy při současném vysílání [18]

3.2 Přenos zpráv

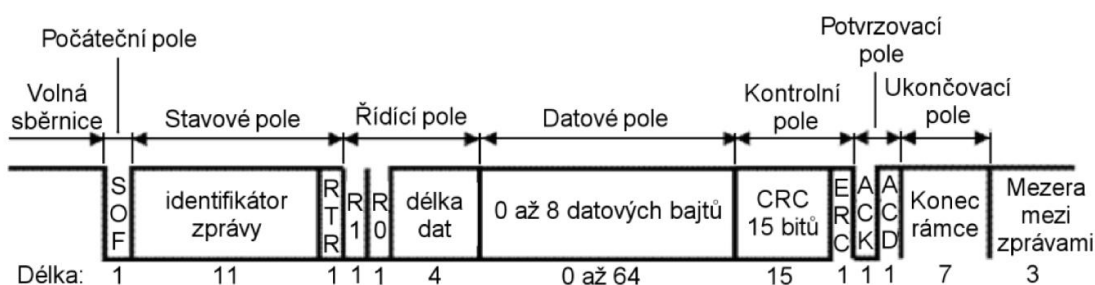
V CAN protokolu existují dva druhy datových zpráv rozlišených podle velikosti identifikátoru. Zpráva s 11bitovým identifikátorem se nazývá **standardní** formát zprávy (Standard Frame), který je definován podle protokolu CAN 2.0A. Specifikace CAN 2.0B definuje zprávu s 29bitovým identifikátorem **rozšířený** formát (Extended Frame). Oba dva typy zpráv mohou být použity na stejné sběrnici, pouze pokud je řadič podporován specifikací CAN 2.0B.

Typy rámců:

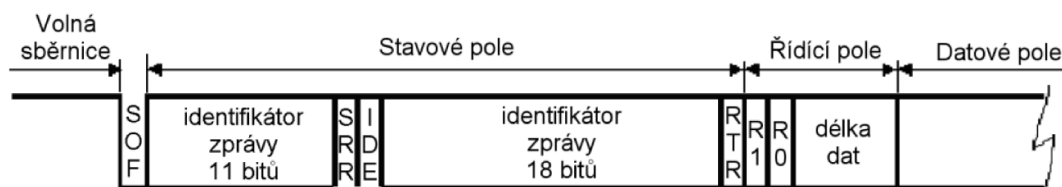
- Data Frame:** Datový rámec může obsahovat datovou zprávu.
- Remote Frame:** Zpráva pro vyžádání odezvy.
- Error Frame:** Zpráva je poslána, pokud jednotka detekuje chybu na sběrnici.
- Overload Frame:** Zpráva o přetížení slouží k oddálení další datové zprávy nebo žádosti o data.

3.2.1 Datový rámec (Data Frame)

Struktura datových rámců je zobrazena na obrázcích číslo 7 a 8. Odeslání datové zprávy je možné, pouze pokud je volná sběrnice. K tomu, aby uzel mohl začít vysílat, musí sledovat stav na sběrnici a pokud detekuje volnou sběrnici, začíná vysílat. Jestli uzel získá kontrolu nad sběrnici, to závisí na použitém mechanismu přístupu k médium.



Obrázek 7: Datová zpráva podle specifikace CAN 2.0A [6]



Obrázek 8: Rozšířená datová zpráva podle specifikace CAN 2.0B [6]

Popis následujících bitů:

SOF:	(Start of Frame) Tento dominantní bit určuje začátek komunikace.
Identifikátor:	(11bitový v případě standardního rámce a 11+18 bitový v případě rozšířeného rámce) udává prioritu a význam zprávy.
RTR:	(Remote Transmission Request Bit) Udává, zda se jedná o datovou zprávu nebo o žádost o přístup na sběrnici. Pro datovou zprávu musí být tento bit v dominantním stavu.
SRR:	(Substitute Remote Request Bit) V rozšířeném formátu má tento bit recesivní hodnotu, SRR je na pozici RTR bitu a proto pokud vznikne kolize rozšířeného a standardního rámce, větší prioritu bude mít standardní rámec.
IDE:	(Identifier Extension Bit) Má vždy recesivní hodnotu.
R1,R2:	Rezervované bity
DLC:	(Data Length Code) Určuje počet přenášených datových bajtů ve zprávě. Může nabývat hodnot od 0 do 8.
Datové pole:	Vlastní data, v závislosti na DLC může být přeneseno maximálně 8 bajtů dat.
CRC:	Kontrolní součet slouží ke zjišťování chyb v přenosu.
ERC:	Oddělovač kontrolního součtu a potvrzení. Vždy v úrovni recessive.
ACK:	Potvrzení přijaté zprávy, informuje o tom, že zpráva byla úspěšně přijata.
ACD:	Oddělovač potvrzení zprávy, vždy v úrovni recessive.
Konec Rámce:	Uzavírá přenášenou zprávu pomocí sedmi recesivními bity.

3.2.2 Žádost o data (Remote Data Frame)

Žádost o data může existovat ve standardním i v rozšířeném rámci. Struktura je podobná jako u datového rámce. Liší se pouze v bitu RTR, který je nastaven do úrovně recessive. Datové pole je nulové a nezáleží na velikosti DLC. Pokud nějaký uzel žádá o zaslání dat, nastaví stejný identifikátor, jako má identifikátor, jehož zprávu požaduje.

3.2.3 Chybový rámeček (Error Frame)

Chybová zpráva slouží k signalizaci chyb na sběrnici a skládá se ze dvou částí:

První část je Error Flags, 6 až 12 dominantních (pro aktivní chybu) nebo recesivních (pro pasivní chybu) bitů. Druhá část chybového rámce je Error Delimiter a ta obsahuje 8 recesivních bitů.

3.2.4 Zpráva o přetížení (Overload Frame)

Používá se, pokud chceme oddálit vysílání následující datové zprávy nebo žádosti o zprávu. Struktura je podobná chybové zprávě. Zpráva o přetížení obsahuje minimálně šest dominantních bitů a následně sedm recesivních bitů. Vysílání může být zahájeno až po konci zprávy, oddělovače chyb, nebo po konci předcházející zprávy o přetížení

3.3 Zabezpečení proti chybám

Nespornou výhodou CAN protokolu je několik mechanismů zabezpečující kontrolu a detekci chyb. Tabulka číslo 2 ukazuje čísla a názvy chyb, které se mohou vyskytnout při komunikaci. Tyto chyby, jež jsou podrobněji popsány níže, mohou být vyčteny ze status registru CAN řadiče.

Vkládání bitu – po pěti následujících shodných bitech je vložen do komunikace bit opačné úrovně. Tato metoda je důležitá pro synchronizování časování. Pokud řadič detekuje více než pět po sobě jdoucích shodných bitů, je vygenerována chyba vkládání bitů (*Stuff Error*).

Kontrola rámce zprávy – některé části rámce mají pevně daný formát, např. CRC oddělovač, ACK potvrzení, konec rámce. Pokud se detekuje jiná hodnota, než se očekávala, detekuje se chyba (*Form Error*).

Potvrzení zprávy – všechny stanice, které v pořádku přijmou zprávu, musí po přijetí zprávy odpovědět dominantním bitem. Vysílací stanice nastaví recesivní úroveň a zároveň čte signál na sběrnici. Pokud žádná stanice nevyšle potvrzení, tak je detekována chyba (*Ack Error*).

Monitoring bitů – vysílací stanice monitoruje stav na sběrnici a porovnává ho s očekávaným stavem. Pokud je zaznamenaný rozdíl, detekuje se Bit chyba (*Bit1Error*, *Bit0Error*).

CRC kód – na konci každého rámce je uveden patnáctibitový kontrolní součet (9). Pokud přijímač detekuje chybný CRC kód, je vygenerována CRC chyba (*CRC Error*).

$$X_{15} + X_{14} + X_{10} + X_8 + X_7 + X_4 + X_3 + 1 \quad (9)$$

Číslo chyby	Jméno chyby
0	No Error
1	Stuff Error
2	Form Error
3	Ack Error
4	Bit1 Error
5	Bit0 Error
6	CRC Error
7	Nevyužito

Tabulka 2: Čísla a význam chyb v LEC (Last Error Code)

3.4 Detekce chyb

V závislosti na chybách se může jednotka nacházet v jednom ze tří níže uvedených stavů:

- Error Active (aktivní chyba)
- Error Pasive (pasivní chyba)
- Bus Off (odpojená sběrnice)

Pokud je jednotka ve stavu **Error Pasive**, může se účastnit komunikace, ale nesmí posílat Active Error Flag. Při detekované chybě jednotka vyšle Error Pasive Flag a následně se čeká na synchronizaci před dalším odesláním. Při stavu **Error Active** a detekované chybě jednotka vyšle Active Error Flag. Jednotka se neúčastní komunikace pouze v případě, pokud je ve stavu **Bus Off**.

Pro detekci chyb jsou v řadiči implementovány dva registry. **TEC (Transmit Error Counter)** pro chyby způsobené odesláním. Chyby způsobené čtením zaznamenává **REC (Receive Error Counter)** registr.

Hodnoty v těchto registrech jsou modifikovány podle určitých pravidel:

1. Pokud přijímač detekuje chybu, hodnota REC registru se zvýší o 1. Toto neplatí, pokud byla detekována chyba Bit Error během posílání Active Error Flag nebo Overload Flag.
2. Když přijímač detekuje dominantní bit jako první bit po odeslání Error Flag, hodnota v REC registru se zvýší o 8.
3. Pokud vysílač pošle Error Flag, hodnota TEC registru se zvýší o 8. Toto neplatí, pokud je vysílač v Error Pasive a detekuje chybu Ack Error, protože nedetekoval dominantní bit kvůli odeslání Error Pasive Flag. Poté se hodnota TEC registru nemění.
4. Jestliže vysílač detekuje Bit Error a zároveň se odesílá Active Error Flag nebo Overload Flag, hodnota TEC registru se zvýší o 8.
5. Když přijímač detekuje Bit error, zatímco se odesílá Active Error Flag nebo Overload Flag, tak se hodnota REC registru zvýší o 8.
6. Každý modul toleruje až sedm po sobě jdoucích dominantních bitů po odeslání Active Error Flag, Pasive Error Flag nebo Overload Flag. Po detekování čtrnácti po sobě jdoucích dominantních bitů, v případě Active Error Flag nebo Overload Flag, nebo osmi po sobě jdoucích bitů v případě Pasive Error Flag vysílač inkrementuje TEC registr o 8 a přijímač inkrementuje REC registr taky o 8.
7. Po úspěšném odeslání zprávy bez chyb se hodnota TEC registru se sníží o 1.
8. Po úspěšném přijmutí zprávy bez chyb se hodnota REC registru sníží o 1, pokud byla v rozsahu 1 až 127. Pokud REC registr byl nulový, tak se

hodnota nemění. Naopak pokud byla hodnota registru větší než 127, tak se hodnota nastaví mezi 119 až 127.

9. Modul přejde do Error Pasive stavu, pokud hodnota TEC nebo REC registru je rovna nebo větší než 128.
10. Modul přejde do stavu Bus Off, pokud je hodnota TEC registru větší než 255.
11. Modul přejde do zpět z Error Pasive do Error Active, pokud jsou obě hodnoty registrů TEC a REC menší než 128.
12. Modul, který je ve stavu Bus Off může přejít do stavu Error Active za předpokladu nulových hodnot TEC a REC registrů. To je docíleno 128 po sobě jdoucích recesivních bitů na sběrnici. [7]

3.5 Programovací model C_CAN

C_CAN je CAN modul používaný v samostatných zařízeních, jako například mikroprocesorech. Je popsán v jazyce VHDL. Obsahuje komponenty, které jsou uvedeny na obrázku 9. Jedná se o CAN jádro, Message RAM, Message Handler, Control Registers a Module Interface.

CAN jádro zpracovává komunikaci podle CAN protokolu verze 2.0 část A a B. Přenosová rychlost může být nastavena až do rychlosti 1 Mbit/s. Pro komunikaci na CAN síti musí být nastaveny individuální message objekty. Tyto message objekty a akceptační filtry jsou uloženy v Message RAM. Všechny funkce týkající se obsluhy zpráv jsou implementovány v Message Handleru. Obsluhou rozumíme filtrování zpráv, přenos zpráv mezi CAN jádrem a Message RAM, generování přerušení a vysílacích požadavků. Přístup CPU do registrů je možný pouze přes module interface.

C_CAN modul alokuje paměťový prostor o velikosti 256 bajtů. Registry jsou šestnáctibitové s tím, že sudá adresa je spodní bajt a lichá adresa patří hornímu bajtu. Aby se zamezilo konfliktům při čtení a zápisu, existují dvě sady interface registrů (IF1, IF2), které slouží k řízení přístupu CPU do Message RAM [2].

3.5.1 CAN registry

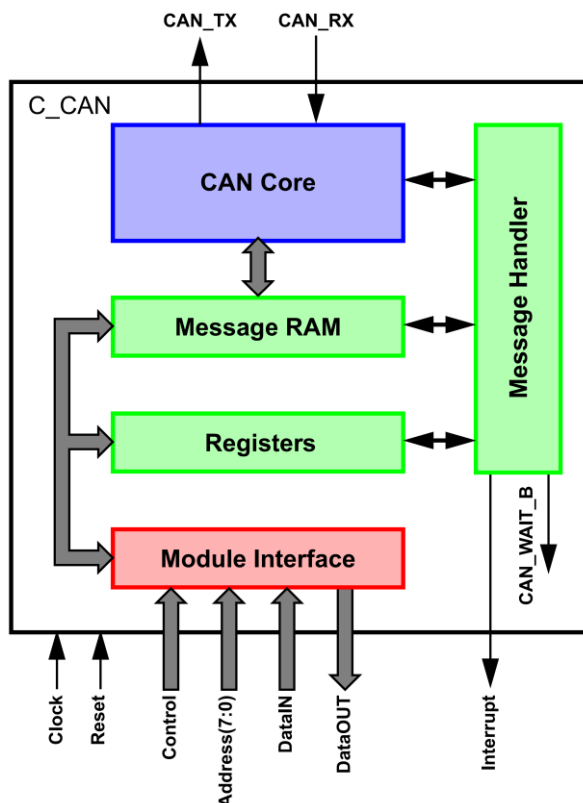
Tabulka číslo 3 ukazuje seznam a adresy všech dostupných registrů. Nevyužité adresy jsou rezervovány pro budoucí rozšíření.

Adresa	Název registru	Adresa	Název registru
0x00	CAN Control Register		
0x02	Status Register		
0x04	Error Counter		
0x06	Bit Timing Register		
0x08	Interrupt Register		
0x0A	Test Register		
0x0C	BRP Extension Register		
0x10	IF1 Command Request	0x40	IF2 Command Request
0x12	IF1 Command Mask	0x42	IF2 Command Mask
0x14	IF1 Mask 1	0x44	IF2 Mask 1
0x16	IF1 Mask 2	0x46	IF2 Mask 2
0x18	IF1 Arbitration 1	0x48	IF2 Arbitration 1
0x1A	IF1 Arbitration 2	0x4A	IF2 Arbitration 2
0x1C	IF1 Message Control	0x4C	IF2 Message Control
0x1E	IF1 Data A 1	0x4E	IF2 Data A 1
0x20	IF1 Data A 2	0x50	IF2 Data A 2
0x22	IF1 Data B 1	0x52	IF2 Data B 1
0x24	IF1 Data B 2	0x54	IF2 Data B 2
0x80	Transmission Request 1		
0x82	Transmission Request 2		
0x90	New Data 1		
0x92	New Data 2		
0xA0	Interrupt Pending 1		
0xA2	Interrupt Pending 2		
0xB0	Message Valid 1		
0xB2	Message Valid 2		

Tabulka 3: Přehled CAN registrů

Vzhledem k počtu všech registrů jsou popisy bitů uvedeny v příloze. Pro inicializaci CAN řadiče se nastaví bit Init z CAN Control registru do logické jedničky. Pro přístup do Bit Timing registru je nutné nastavit bit CCE. Zajištění normální operace spustíme smazáním bitu Init. Vlivem chyb na sběrnici se řadič může dostat do stavu Bus Off, v němž se mu automaticky nastaví bit Init do logické jedničky a pro restartování CAN řadiče je nutné smazat hodnotu bitu Init. CAN řadič vyčká na 129 výskytů jedenácti po sobě jdoucích recesivních bitů (stav Bus Idle) a teprve poté pokračuje v normální operaci. Během obnovení inicializace monitoruje řadič stav na sběrnici, a pokud místo

sekvence jedenácti recesivních bitů přečte bit dominantní, řadič zaznamená chybu Bit0Error.



Obrázek 9: Blokové schéma C_CAN modulu [2]

3.6 Time-triggered CAN (TTCAN)

Nový koncept elektronicky řízených automobilových systémů vyžaduje komunikaci řízenou na základě časových událostí. Jedná se zejména o systémy X-by-Wire, které nahrazují mechanické a hydraulické systémy doposud používané k ovládání akcelerační, brzdění a řízení. Cílem TTCAN je odstranění časové prodlevy při současném požadavku pro vysílání více stanic a zavedení determinismu na sběrnici. Toho se dosáhne spuštěním periodické komunikace, která je taktovaná referenční zprávou od řídicí stanice. TTCAN rozšiřuje původní protokol o možnost TDMA přístup, ale nechává vyhrazenou část pro původní nedeterministický přístup. Více informací lze nalézt v normě ISO 11898-4. [20]

4 KONCEPCE

Cílem návrhu bylo vytvořit aplikační moduly, které budou schopné komunikace po CAN protokolu. Tyto moduly bude možné propojit do sběrnice pomocí CAN kabelu. Libovolný modul se může připojit k PC pomocí USB. Aktuálně připojený modul bude pomocí příkazů z PC řídit komunikaci nebo měnit vlastnosti sběrnice. Bude také možné vyčíst stav sběrnice, parametry CAN řadiče a chyby vzniklé při komunikaci s moduly.

Návrh modulů vychází z předchozích verzí CAN Application Board V.1 a V.2 [12] ve firmě ON Semiconductor. Moduly navržené v této práci jsou kompaktní, využívají USB připojení k počítači a skládají se jen z nezbytných periférií. Vytvořili jsme dva druhy modulů lišící se ve využití cívky souhlasného napětí. Modul s označením CAN Application Board V3a obsahuje cívku souhlasného napětí. Snaha o co nejmenší rozměry vedla k vytvoření dalšího modulu s názvem CAN Application Board V3b. Tento modul neobsahuje cívku souhlasného napětí a rozměry desky jsou o 12,5 mm užší než v případě modulu s cívkou. Při výrobě desek plošných spojů jsme využili metodu panelování na jeden přířez, tudíž jsme si mohli dovolit vyrobit dva druhy desek za cenu jednoho návrhu.

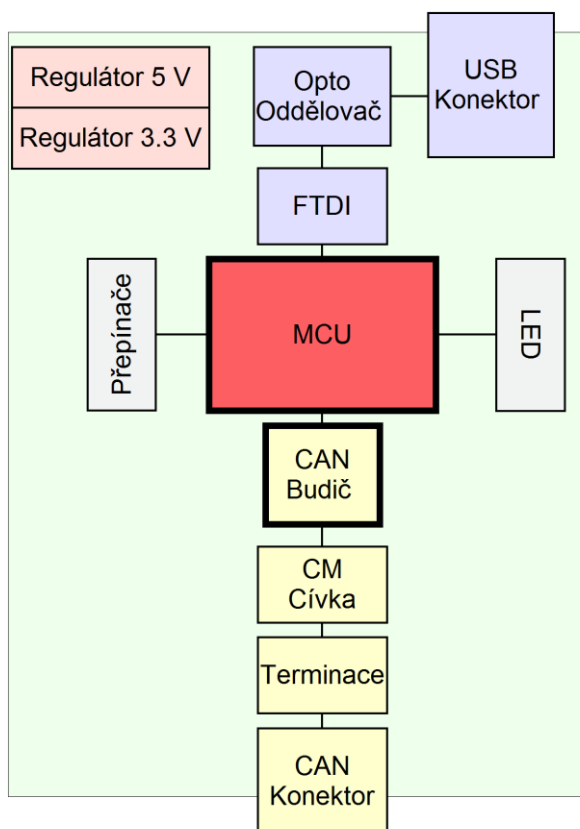
Moduly obsahují pouze jednu CAN sběrnici. Při vývoji jsme zamýšleli využít druhou pomocnou sběrnici pro řízení modulů, ale tato varianta byla zamítnuta z několika důvodů. Tato CAN síť byla navržena také pro testování v EMC komoře, a tam by pomocná sběrnice mohla působit problémy, jako např. rušení samotných modulů. Další podstatný problém by byl s velikostí modulů a nutností použití externí kabeláže, proto bylo rozhodnuto o využití pouze jedné CAN sběrnice. Nicméně pro další možné rozšíření jsme vyvedli osmi pinový konektor.

Komunikace po sběrnici je řešena pomocí datových zpráv, kdy první bajt určuje příkaz a ve zbylé části zprávy se nachází parametry. Každý modul poslouchá komunikaci na sběrnici a zaznamenává případné chyby. Tyto chyby lze později načíst do počítače. Každá chyba je také signalizována pomocí informačních LED diod.

Mikroprocesor v modulech kontroluje a zpracovává příchozí zprávy ze sériové linky anebo čeká na přerušení od CAN řadiče a zpracovává CAN příchozí zprávu. Firmware je pro všechny moduly stejný. Moduly se odlišují pouze přepínačem, který určuje jejich adresu.

5 NÁVRH HARDWARE

Požadavky na novou verzi modulu byly následující: snaha o co nejmenší rozměry, použít pokud možno polovodičové součástky vyráběné přímo firmou ON Semiconductor, místo sériové linky začlenit převodník UART-USB od firmy FTDI a využít galvanického oddělení počítače a modulu pomocí optočlenů.



Obrázek 10: Blokové schéma modulu

Blokové schéma je uvedeno na obrázku číslo 10. Hlavní řídicí jednotkou je mikroprocesor, který zpracovává příkazy z PC a monitoruje nebo aktivně řídí provoz na sběrnici. CAN budič převádí signál z mikroprocesoru na signál pro CAN sběrnici.

5.1 Mikroprocesor

Mikroprocesor bude řídit CAN komunikaci modulu a komunikovat s počítačem přes UART sběrnici. Při vývoji je k dispozici developerské prostředí KEIL μ Vision 2 s USB Debug Adapter programátorem pro C8051Fxxx mikroprocesory. Kódové označení tohoto programátoru firmy Silicon Laboratories je DEBUGADPTR1-USB.

Požadavky na mikroprocesor byli následující:

- podpora CAN protokolu
- jednoduchá obsluha
- dostupnost na trhu
- dostupné developerské nástroje
- softwarová podpora

Na trhu se nachází velké množství mikroprocesorů s podporou CAN řadiče.

Dostupné mikroprocesory s CAN řadičem		Poznámka
Atmel	AT89C51CC03CA	Velká paměť programu, podpora prostředí KEIL.
Freescale	MC9S08*	Velký výběr MCU, nízká cena.
Infenion	SAK-C505CA	Velmi drahý mikroprocesor.
NXP	P87C591	Běží na jádře 8051. Je drahý.
Microchip	PIC18F258-I	Málo kusů na skladech v internetových obchodech.
Silicon Labs	C8051F04x	Kvalitní podpora pro CAN protokol. Velká paměť programu, velká RAM.
ST	STM8S207xx	Malá softwarová podpora pro CAN protokol.

Tabulka 4: Výčet dostupných mikroprocesorů s CAN řadičem

V tabulce číslo 4 je uvedeno několik zástupců mikroprocesorů různých výrobců. Při výběru byl kladen důraz na kvalitu podpory ovladačů, zdrojových kódů a dostupné vývojové prostředí. Vzhledem k dostupné licenci pro prostředí KEIL a velké podpoře zdrojových kódů pro mikroprocesory řady C8051F0xx od společnosti Silicon Laboratories byl zvolen mikroprocesor C8051F044 s 64 kB paměti programu. Paměť RAM má velikost 4.25 kB. Mikroprocesor funguje na jádře 8051. Frekvence externího krystalu byla zvolena na 8 MHz z důvodu zajištění přesného časování CAN sběrnice. Velikost T_q se rovná 125 ns. Pro rychlost 1 Mbit je nutné, aby měl Bit time velikost osmi T_q .

Podle katalogového listu má krystal X1 řady 12SMX kapacitní zátěž $C_L = 16$ pF. Podle vzorce 10 určíme velikost paralelní kapacity za předpokladu, že C_{L1} se rovná C_{L2} a velikost C_{STRAY} je 3 pF podle katalogového listu mikroprocesoru. C_{STRAY} určuje vstupní kapacitu mezi piny XTAL1 a XTAL2 mikroprocesoru.[15]

$$C_L = \frac{C_{L1} \cdot C_{L2}}{C_{L1} + C_{L2}} + C_{STRAY} [pF] \quad (10)$$

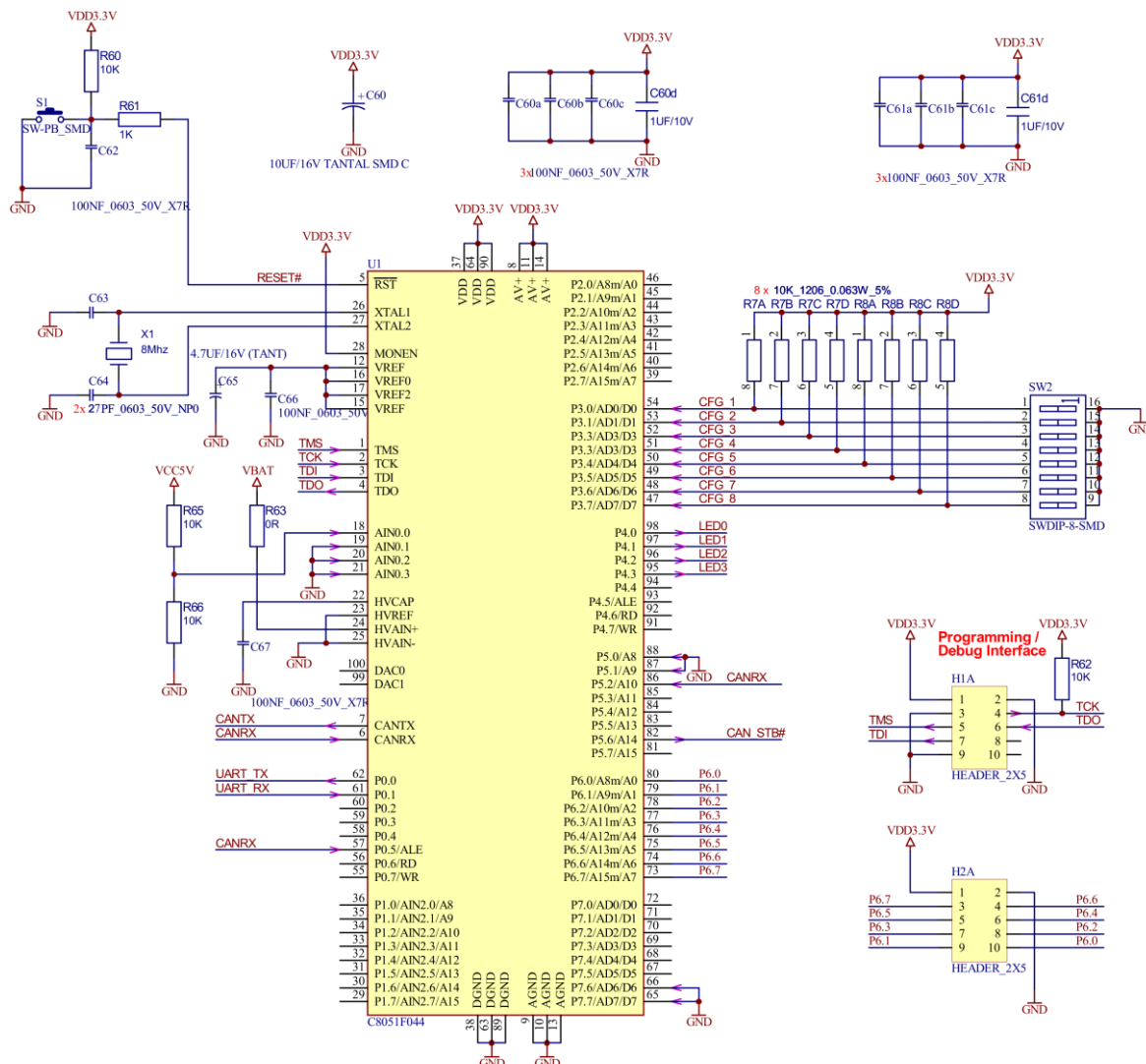
Po úpravě dostaneme následující vztah:

$$C_{L1} = 2 * C_L - 2 * C_{STRAY} [pF] \quad (11)$$

$$C_{L1} = 2 * 16 - 2 * 3 pF$$

$$C_{L1} = 26 \text{ pF}$$

S ohledem na dostupnou výrobní řadu kondenzátorů E12 jsme zvolili pro stabilizační kondenzátory C63 a C64 velikost kapacity 27 pF.



Obrázek 11: Schéma MCU

Programovací konektor H2A slouží k připojení programátoru / debuggeru. Mikroprocesor využívá JTAG protokolu, popis pinů je uveden v tabulce číslo 5.

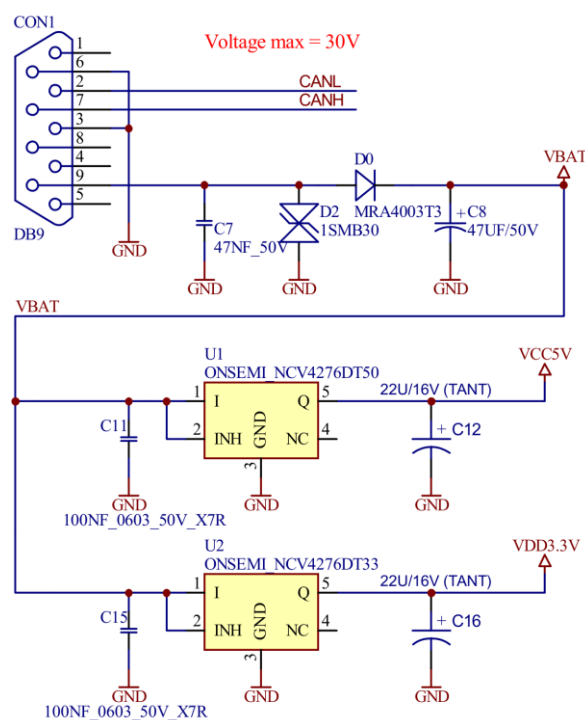
Pin konektoru	Pin mikroprocesoru	Popis	Pin konektoru	Pin mikroprocesoru	Popis
8, 10	-	Nezapojeno	6	4	TDO
2,3,9	38, 63, 89	Zem	7	3	TDI
4	2	TCK	1	37, 64, 90	Napájení
5	1	TMS			

Tabulka 5: Popis pinů JTAG konektoru

$$R_P = \frac{3.3 - 1.9}{0.003} \Omega$$

$$R_P = 466 \Omega$$

Jako napěťové regulátory byly použity



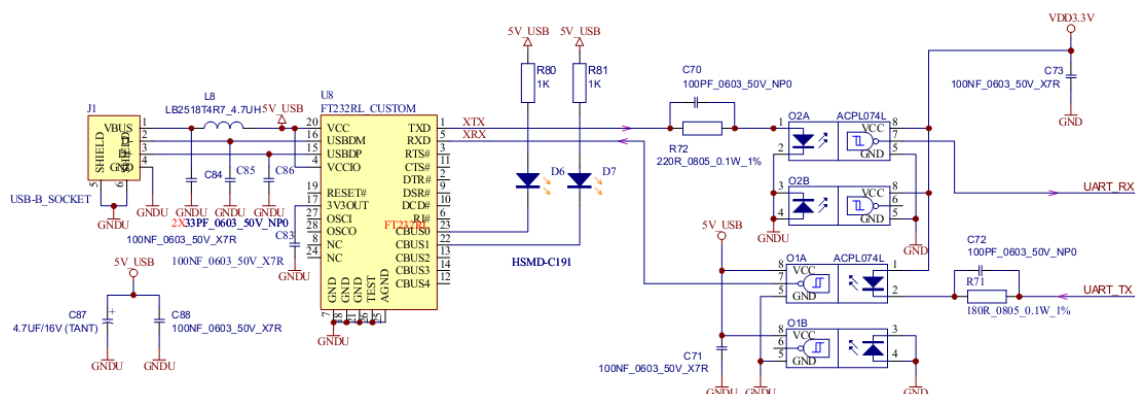
listu začíná na 35 V. Tato dioda je tedy vhodná pro ochranu regulátorů a stabilizačních kondenzátorů, které mají maximální provozní napětí 50 V. V tabulce číslo 6 je popsána teoretická katalogová maximální proudová spotřeba všech bloků. Předpokládaná maximální spotřeba modulu je 280 mA. Nicméně tato hodnota je vyčtena z maximálních hodnot katalogových listů a ve skutečnosti je spotřeba modulu nižší.

Maximální proudová spotřeba	5 V větev	3.3 V větev
MCU	-	150,0 mA
CAN budič	75,0 mA	-
Pull Up – MCU	0,3 mA	3,2 mA
CAN sběrnice	35,0 mA	-
LED diody	3,0 mA	13,8 mA
Celkem	113,3 mA	167,0 mA

Tabulka 6: Maximální katalogová proudová spotřeba

5.3 USB-UART Interface

Pro komunikaci s počítačem je použit jednočipový převodník z USB na UART s označením FT232RL a s oddělovacími optickými členy ACPL074L. Tento optický člen má na vstupu vysokorychlostní LED diodu a na výstupu CMOS detektor v integrovaném obvodu. K tomuto integrovanému obvodu je nutné připojit napájení. Na obrázku číslo 13 je zobrazeno schéma. Diskrétní součástky pro obvod FT232RL, který je ve schématu označen jako U₈, jsou zvoleny podle doporučeného katalogového zapojení. Dále jsem zvolil dvě signalizační LED diody pro zobrazení funkčnosti komunikace. Vstup *UART_TX* signálu do optického členu je limitován rezistorem s paralelním kondenzátorem, který slouží k urychlení hran signálu. Digitální výstup z optického členu pro signál *UART_RX* je veden do mikroprocesoru. Kondenzátory C73 a C71 respektive C87 a C88 slouží jako blokovací.

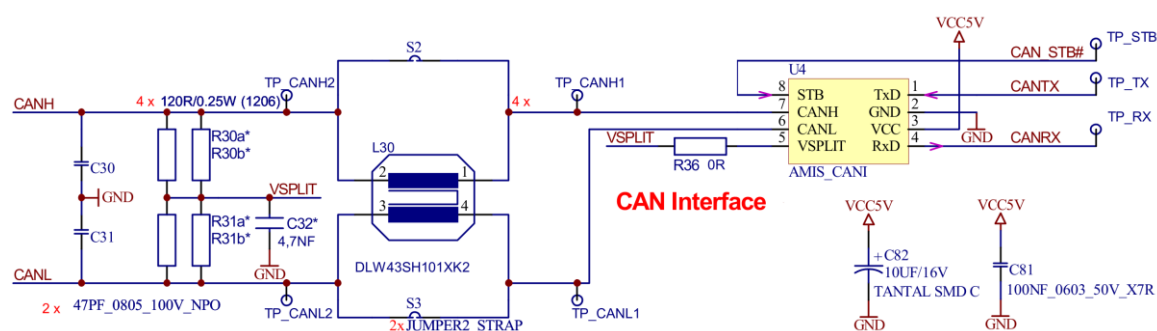


Obrázek 13: Schéma zapojení FT232RL převodníku

5.4 CAN budič

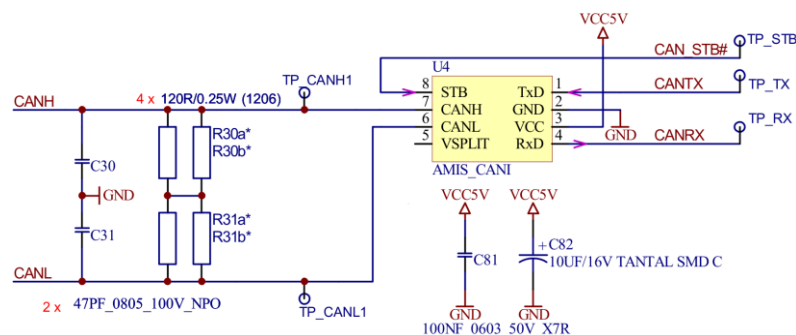
Pro správnou funkci CAN sítě je třeba použít CAN budič, který slouží jako převodník mezi mikroprocesorem a sběrnici CAN. Protože tato deska je zamýšlená jako testovací, můžeme zde zapojit libovolný osmi pinový budič, jako např. NCV7340, NCV7351 nebo NCV7342 či podobný čip se stejně rozmístěnými piny. U každého pinu je vyveden testovací bod, na který se může připojit osciloskop, nebo logický analyzátor. Na straně sběrnice je možnost připojit zakončovací rezistory, nebo cívku souhlasného napětí. Také lze volitelně připojit kondenzátor C32, který podporuje stabilizaci napětí na sběrnici. Kondenzátory C30 a C31 slouží jako přídavná ochrana proti ESD. Tyto kondenzátory nejsou standardně osazeny. Zakončovací rezistory R30 a R31, ve velikosti pouzdra 1206, jsou zapojeny paralelně z důvodu vyšší výkonové ztráty. Celkový stejnosměrný výkon procházející zakončovacími rezistory může být až 0.5 W.

Některé budiče, jako např. NCV7340, mají výstupní pin Vsplrit, který generuje napětí určené pro stabilizaci sběrnice.[13] Pokud by bylo potřeba toto propojení odpojit, je k dispozici nulový rezistor R₃₆. U některých typů budičů není nutné zapojovat Vsplrit pin. Například obvod NCV7351 má na vývodu číslo 5 výstupní napětí z regulátoru pro použití v aplikaci.



Obrázek 14: Schéma zapojení CAN budiče s cívkou

Pro testování čipů bez cívky souhlasného napětí jsme vytvořili následující modifikaci uvedenou na obrázku číslo 15. Výsledkem této úpravy jsou menší rozměry desky plošných spojů a kratší vedení cest sběrnice od budiče ke konektoru.



5.5 CAN konektor

Konektor je zapojen podle specifikace [4]. Použili jsme typický 9pinový D-Sub konektor typu samec. Tento konektor obsahuje dvojici CAN signálů CANH a CANL. Pro napájení lze použít devátý pin konektoru. Třetí a šestý pin je zemnicí. Volitelný pin pro CAN zemnění jsme nezapojovali, protože jsme použili CAN kabel od firmy Draka DE a tento kabel nemá stínění. Zapojení konektoru lze vidět ve schématu na obrázku číslo 12.

Pin #	Signál	Popis	Zapojení
1	-	Rezervován	Nezapojeno
2	CAN_L	CANL signál na sběrnici	CANL
3	CAN_GND	CAN Ground	GND
4	-	Rezervován	Nezapojeno
5	CAN_SHLD	Volitelné CAN stínění	Nezapojeno
6	GND	Volitelné CAN zemnění	GND
7	CAN_H	CANH signál na sběrnici	CANH
8	-	Rezervován	Nezapojeno
9	CAN_V+	Napájení pro aplikaci	VBAT

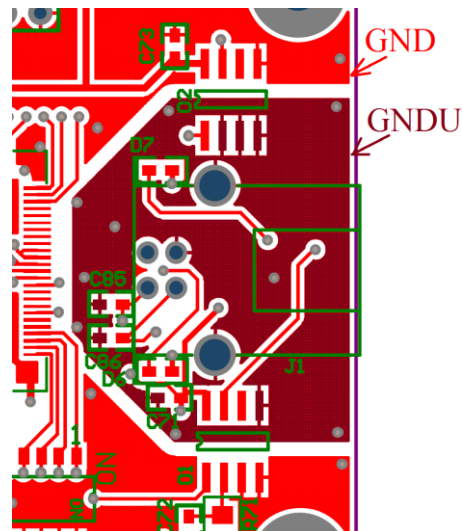
Tabulka 7: Zapojení pinů propojovacího konektoru

6 NÁVRH MODULU

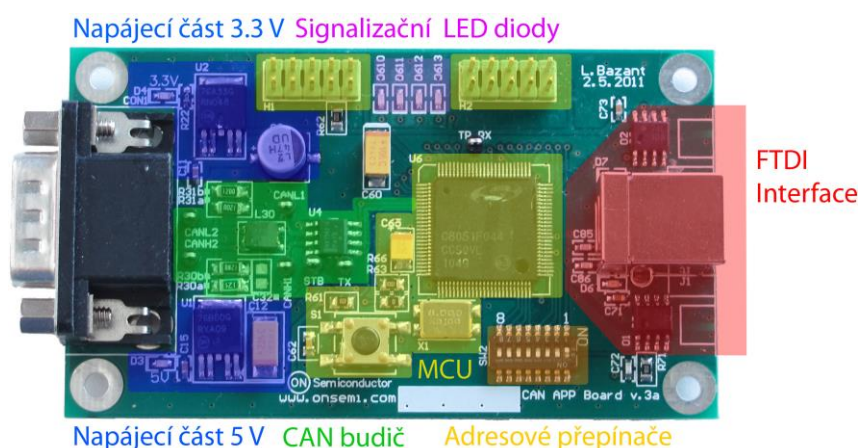
Design jsme prováděli v návrhovém systému Protel 99. Podařilo se navrhnout oboustrannou desku plošných spojů o rozměrech 50 x 86 mm pro verzi s cívkou souhlasného napětí. Druhý návrh bez cívky souhlasného napětí měl velikost 50 x 74.5 mm. Byl kladen důraz na správné návrhové zásady, především na co nejkratší a symetricky vedené vodiče CANH a CANL. Dále zapojení blokovacích kondenzátorů blízko napájecím pinům mikroprocesoru a budiče. Pouzdra všech rezistorů kromě pull-up rezistorů pro přepínač a rezistorů na CAN sběrnici jsme zvolili ve velikosti 0805. Pro použití zakončovacích rezistorů na CAN sběrnici jsme použili kvůli vyššímu proudovému výkonu rezistory s velikostí pouzdra 1206. Pro 100 nF blokovací kondenzátory jsme zvolili velikost pouzdra velikosti 0603.

Nejmenší šířka signálových spojů je ve velikosti 10 mils. Pro napájecí část jsme zvolili minimální šířku spojů 40 mils. Také pro spoje na CAN sběrnici jsme zdvojnásobili šířku cest.

Pro Galvanické oddělení mikroprocesoru a USB jsme navrhli dvě zemní vrstvy (GND a GNDU) a oddělili jsme je mezerou. Napěťové signály Tx a Rx z mikroprocesoru jsou vedeny na FTDI převodník přes optočleny. V designu jsou vedeny dvě napájecí větve, které jsme se snažili nakreslit pokud možno co nejpříměji a vyvarovat se křížení s datovými cestami a s CAN sběrnici.



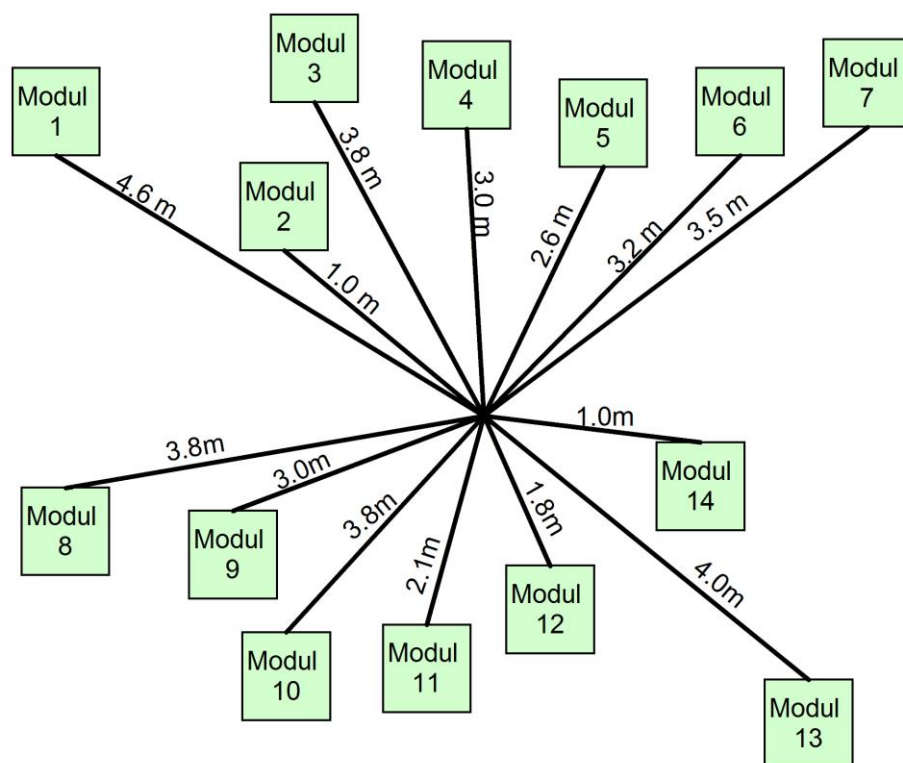
Obrázek 16: Ukázka galvanického oddělení



Obrázek 17: Blokové zobrazení modulu

6.1 Propojení modulů

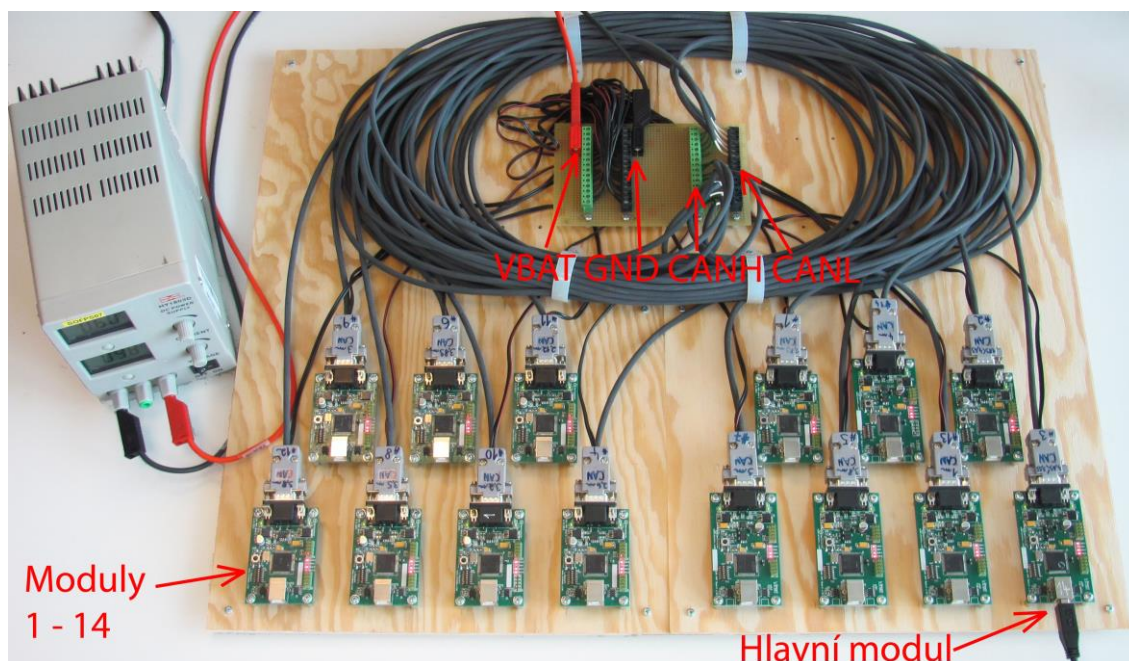
Jako způsob propojení modulů jsme zvolili zapojení do hvězdy podle obrázku číslo 18. Délky propojovacích kabelů byly vybrány podle požadavků firmy ON Semiconductor, což je část reálného způsobu zapojení modulů používaných v osobních automobilech. Zakončovací rezistory jsou zapojeny ve dvou nejvzdálenějších modulech, a to v modulu číslo 1 a 13.



Obrázek 18: Zapojení modulů do sítě

Moduly bylo nutné umístit na podložku. Jako materiál byla zvolena překližka o tloušťce 5 mm a rozměrech 67 x 52 mm. Vyvrtali jsme otvory pro uchycení modulů a pro pomocnou desku. Propojení mezi překližkou a moduly je řešeno pomocí distančních sloupků o délce 12 mm. Pomocná deska z univerzálního plošného spoje obsahuje jen svorkovnice, do kterých je připojeno napájení a sběrníkový kabel. Na této desce jsou také z důvodu přehlednosti nalepeny poznámky, co která svorkovnice znamená.

Černá a červená zdířka slouží k přivedení napájení. Jeden modul má klidovou spotřebu 33 mA. Maximální naměřená spotřeba činí 53 mA. Při zapojení všech 14 modulů potřebujeme 12 V zdroj s výstupním proudem minimálně 900 mA.



Obrázek 19: Zapojení modulů

6.2 LED indikace

Pro jednoduché zobrazení různých stavů je využito LED diod. Po resetu proběhne inicializace MCU, což značí rozsvícení LED1 a LED3. Jakmile se mikroprocesor dostane do normálního stavu, LED diody zhasnou. Po úspěšném skenování sítě všechny moduly rozsvítí LED0. Pokud nastanou chyby, buď při příjmu, nebo při odesílání, je to signalizováno podle obrázku číslo 20.

	LED0	LED1	LED2	LED3
Inicializace MCU	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Normální stav	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Skenování sítě	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Chyba příjmu	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Chyba odesílání	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Obrázek 20: Stavy LED diod

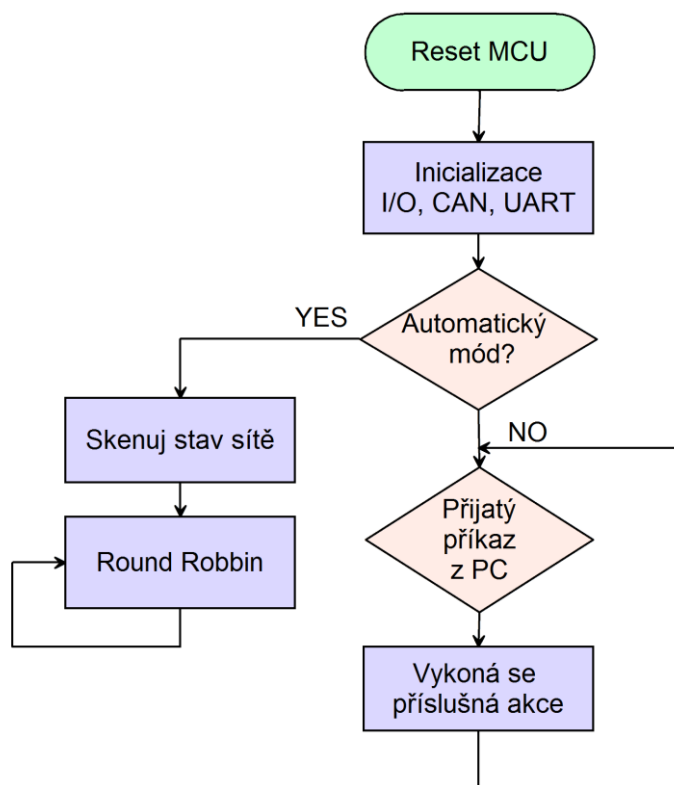
7 SOFTWARE MODULU

7.1 Software pro MCU

Firmware, software pro mikroprocesor, byl napsán ve vývojovém prostředí KEIL μ Vision 2 v jazyku C. Vývojový diagram hlavního programu *main.c* je zobrazen na obrázku číslo 21. Po resetu MCU se inicializují periferie a zapne se přerušení. V závislosti na osmém pinu adresového přepínače se MCU může dostat do automatického módu, kdy nepotřebujeme počítač pro ovládání, ale MCU automaticky skenuje síť a začne posílat zprávy ostatním uzlům na sběrnici.

Pokud není nastavený automatický mód, čeká se na přijmutí příkazu z PC. Podle typu příkazu se vykoná příslušná akce, jako např. zjištění aktuální adresy modulu, odeslání datové zprávy na CAN sběrnici nebo změna časování sběrnice. Popis všech podporovaných příkazů je popsán v tabulce číslo 9.

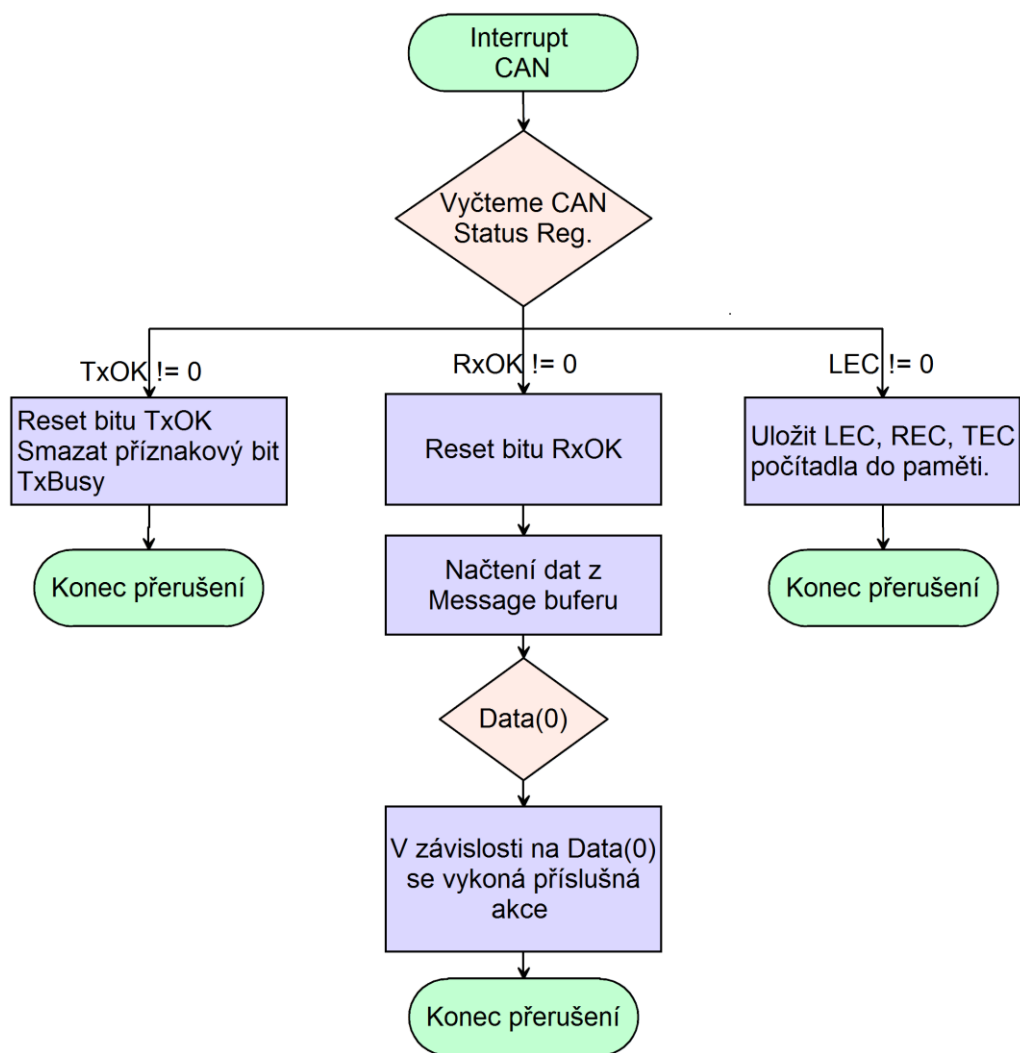
Inicializace CAN řadiče probíhá tak, že nejprve se smaže všech 32 message objektů. Následně se v každém modulu inicializuje jeden přijímací message objekt a jedenáct odesílacích. Pro každou velikost datové zprávy jsem vytvořil samostatný message objekt a další dva na odesílání příkazů. Následuje konfigurace CAN Control Registru, kdy je nutné nastavit konfigurační bity Init a CCE do logické jedničky a až teprve poté se může měnit další nastavení CAN registrů, jako je např. časování, command mask registr apod. Konfiguraci ukončíme smazáním konfiguračních bitů Init a CCE. Dále povolíme přerušení a smažeme stavové proměnné.



Obrázek 21: Vývojový diagram programu main.c

CAN řadič je nastaven tak, aby zpracovával všechna přerušení z CAN sběrnice. Pokud nastane přerušení s číslem 19, což znamená, že se jedná o přerušení z CAN řadiče, MCU vyčte status registr. Dále se zjistí, co bylo příčinou tohoto přerušení. Pokud je bit TxOK nastaven do logické jedničky, znamená to, že došlo k odeslání CAN zprávy. Ručně se vynuluje TxOK bit a smaže se příznakový bit Txbusy, který indikuje, zda byla zpráva odeslána či nikoliv. Pokud se vyčte nenulová hodnota LEC bitů, znamená to, že řadič detekoval chybu. Proměnné LEC, REC a TEC se uloží do paměti pro možnost pozdějšího vyčtení.

Příjem zprávy je detekován nenulovou hodnotou RxOK bitu. Následně je tento bit smazán. MCU přečte data z přijaté zprávy. První bajt z této zprávy udává akci, kterou musí MCU vykonat jako například změna stavu LED diody, odeslání zprávy nazpět hlavnímu modulu, nebo změna CAN časování. Seznam všech podporovaných příkazů pro CAN přerušení je uveden v tabulce číslo 8.



Obrázek 22: Vývojový diagram pro CAN přerušení

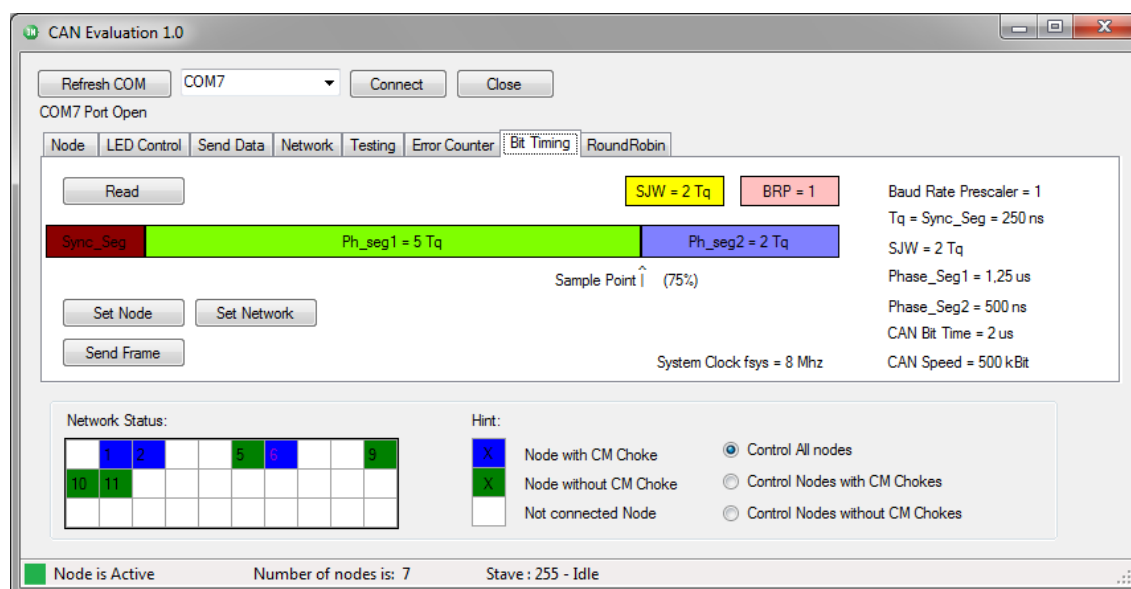
Funkce	Data(0)	Data(1)	Data(2)	Data(3)	Poznámka
Odeslání Odezvy	'N'	-	-	-	Odešle zprávu s adresou modulu
Přijmutí odezvy	'n'	Adresa	-	-	Master si uloží adresu modulu do paměti
Odeslání testu	'T'	-	-	-	Odešle testovací zprávu
Přijmutí testu	't'	Adresa	-	-	Přijme testovací zprávu
Kontinuální test	'I'	-	-	-	Odeslání testovacích zpráv
Kontinuální test	'i'	adresa	-	-	Přijmutí a zpracování testovacích zpráv
Změna časování	'a'	Tseg1	Tseg2	SJW	BRP,
Led	'l'	Adresa	Hodnota	Styl	Změna hodnoty LED
Změna módu	'D'	'a' / 'w'	-	-	'a' = arbitrace, 'w'=bez arb.

Tabulka 8: Seznam všech podporovaných příkazů pro CAN přerušení

7.2 Ovládací software pro PC

Software pro PC je napsán v jazyce Visual Basic. Vytvořeny jsou následující funkce:

- vyčtení adresy právě připojeného modulu
- ovládání stavových LED diod
- odeslání volitelného datového rámce
- zjištění počtu modulů v síti a grafické znázornění
- testování komunikace mezi moduly
- vyčtení aktuálního chybového registru
- čtení chyb, které nastaly v průběhu komunikace
- uložení chyb do csv souboru
- nastavení časování sběrnice včetně změny samplovacího bodu
- vizuální testování modulů
- vyčtení a nastavování status registrů



Obrázek 23: Ovládací program pro CAN moduly

7.3 Popis programu

Po spuštění programu CAN Evaluation se musí spojit COM port s připojeným fyzickým modulem. Tlačítko Refresh COM port nám zobrazí všechny dostupné COM porty na aktuálním počítači. Vybereme správný COM port a po stisknutí tlačítka Connect se automaticky připojí na aktuální modul. Pokud je modul v pořádku, podaří se vyčíst jeho adresa. Tím že jsme se propojili s jedním modulem, stává se Masterem, který řídí komunikaci v síti.

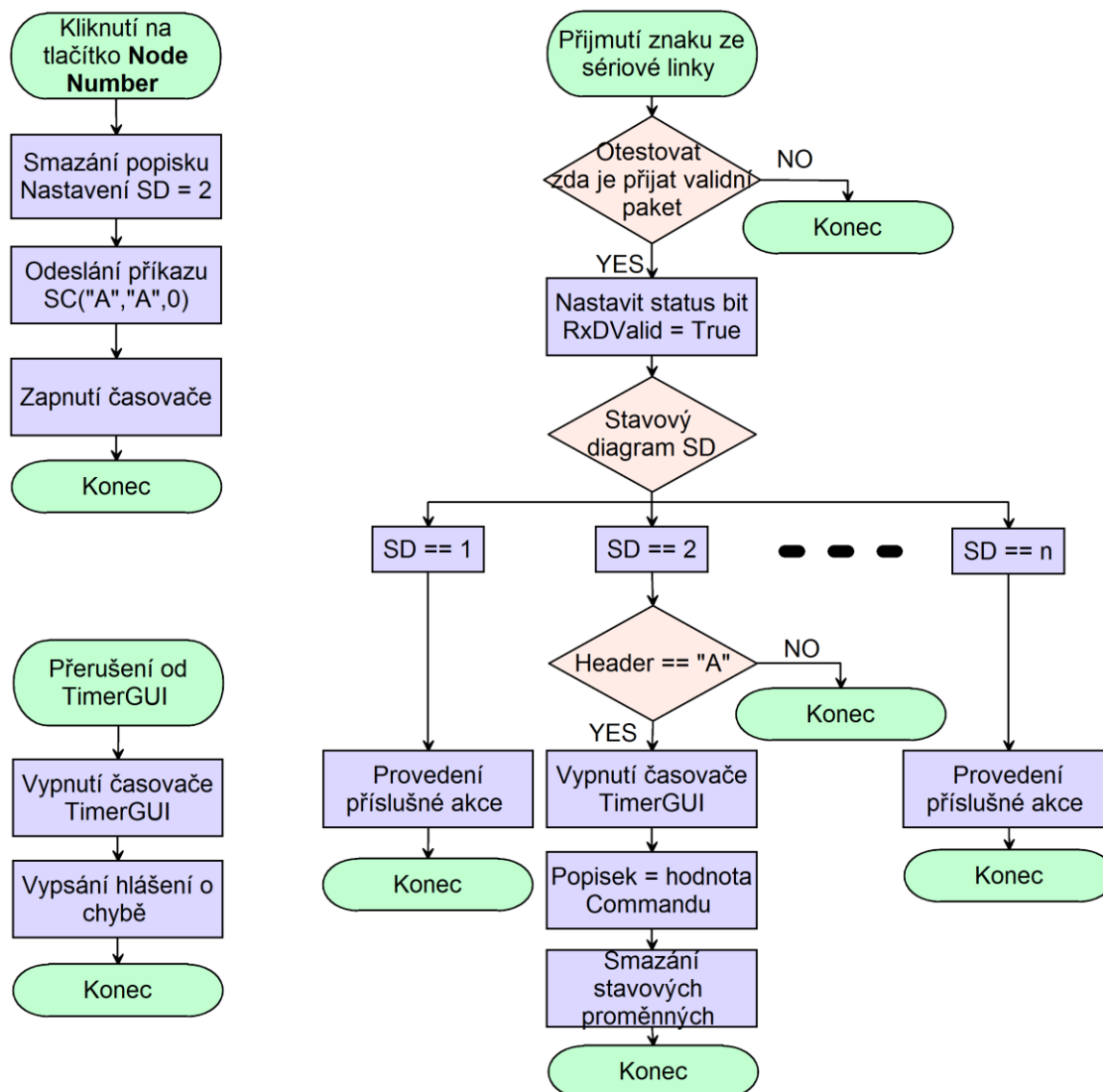
7.4 Vnitřní struktura programu

Program se skládá ze standardních uživatelských ovládacích prvků. Využíváme přerušení pro čtení nových dat ze sériové linky. Pro definování akce je použito stavového diagramu, který se stará o provedení příslušné akce. Časovač TimerGUI slouží k zobrazení chyb, pokud modul neodpoví do požadovaného časového intervalu.

V případě, že chceme např. vyčíst adresu aktuálního modulu, klikneme na tlačítko Node Number, smaže se původní popis, poté se nastaví ukazatel na stavový diagram SD = 2. Sériovou linkou se odešle příkaz „A“, na který by měl mikroprocesor odpovědět posláním aktuální adresy a zapne se časovač TimerGUI s intervalem 250 ms. Pokud do 250 ms nedostane program odpověď od MCU, vypíše se chybové hlášení.

Pro přístup k sériové lince je použita komponenta SerialPort. Pomocí přerušení jsou vytčena data ze sériového kanálu a ihned je provedena kontrola, zda přijatý paket není v rozporu s pravidly uvedenými v následující kapitole. Pokud je přijat správný paket, nastaví se příznakový bit RxDValid a vynuluje se pointer ukazující na aktuální přijatý bajt v paketu.

Po přijetí validního paketu se vykoná akce podle ukazatele na stavový diagram. Otestuje se, zda odpovídá hlavička paketu očekávané hodnotě. Pokud ano, tak se vypíše hodnota commandu, která v sobě nese informaci o aktuální adrese modulu. Po vykonání všech operací se vynuluje ukazatel na stavový diagram, zastaví se časovač a nastaví se správný vzhled komponent. Vývojový diagram pro zjištění aktuální adresy modulu je vyobrazen na obrázku 24.



Obrázek 24: Vývojový diagram pro zjištění adresy aktuálního modulu

7.5 Komunikace mezi PC a modulem

Moduly jsou k PC připojeny pomocí USB kabelu. Převodník FT232RL z USB na UART vytvoří emulovaný sériový port, na který můžeme jednoduše přistupovat. Rychlost komunikace je nastavena na 9600 baud/s při použití jednoho stop bitu, osmi datovými bity, nulové parity a bez použití hardwarové řízení komunikace.

Komunikační protokol spočívá v odeslání a přijímání konkrétního paketu, jenž má definovanou strukturu podle obrázku 25. Celkově může mít paket velikost od 5 bajtů do 13 bajtů.

Délka paketu 1 bajt	Header 1 bajt	Command 1 bajt	Data 0 - 8 bajtů	CRC 1 bajt	Znak 0x0D 1 bajt
------------------------	------------------	-------------------	---------------------	---------------	---------------------

Obrázek 25: Ukázka uspořádání jednotlivých bajtů v paketu

- Délka paketu** – udává celkovou délku paketu. Může nabývat hodnot od 5 do 13. Pomocí tohoto bajtu se dokáže určit, kolik bude přeneseno datových bajtů odečtením hodnoty bajtu minus 5.
- Header** – hlavička paketu obsahuje identifikátor, který určuje, co se má vykonat.
- Command** – pomocný příkaz, který slouží k posílání informací.
- Data** – v paketu může být volitelný počet datových bajtů v závislosti na délce paketu. Maximální povolený počet datových bajtů je 8.
- CRC** – kontrolní součet minimalizuje riziko špatného přenosu paketu.
- Znak 0x0D** – zakončovací bajt má fixní hodnotu 0x0D, po přijmutí tohoto posledního bajtu MCU nebo PC.

Komunikační paket se skládá z informačního bajtu o celkové délce celého paketu, z tohoto bajtu se pozná, kolik datových bajtů je zrovna přenášeno. Následuje Hlavička a Command. Poté může odeslat až 8 datových bajtů. CRC se zaměřuje na testování, zda se paket odeslal správně a na poslední pozici musí být ukončovací bajt o hodnotě 0x0D.

Pokud program, jak PC nebo Firmware, detekuje špatný paket, tento paket je ignorován a nepracovává se.

Funkce	Header	Command	Data	Poznámka
Adresa modulu	'A'	'A'	-	Modul odešle adresu
Echo	'E'	'E'	-	Modul odešle Echo
Odeslání rámce	'X'	'X'	-	
Ovládání LED	'L'	0 _b 0000L ₃ L ₂ L ₁ L ₀	Modul	Nastavení jednoho konkrétního modulu
Ovládání LED	'I'	0 _b 0000L ₃ L ₂ L ₁ L ₀	-	Nastavení všech modulů
Změna módu	'M'	MODE	-	MODE = {0, 1, 2} {normál, cívka, bez c.}
Datová zpráva	'S'	Počet bajtů	Data	Odešle datový rámec s volitelnými bajty
Seznam modulů	'N'	'N'	-	
Test 1 cyklus	'T'	'T'	-	
Test N cyklů arbitrace	'I'	'a'	N	Data ₀ = N _H ; Data ₁ = N _L ; N je (1-65535) Testování s arbitrací
Test N cyklů bez arbitrace	'I'	'w'	N	Data ₀ = N _H ; Data ₁ = N _L ; N je (1-65535) Testování bez arbitrace
Vyčtení chyb	'R'	'R'	-	Čtení aktuálních hodnot z Error registru
Čtení více chyb	'O'	'O'	-	Požadavek na vyčtení pole chyb.
Časování modulu	'B'	'N'	4	Tseg ₂ , Tseg ₁ , SJW, BRP
Časování sítě	'B'	'A'	4	Tseg ₂ , Tseg ₁ , SJW, BRP
Čtení registrů	'G'	'G'	-	Modul odešle informace o CAN reg.
Round Robin	'r'	'D'	-	Vizuální testování celé sítě. První mód
Round Robin	'r'	'M'	-	Vizuální testování celé sítě. Druhý mód
Změna rychlosti	's'	's'	2	Rychlost Round Robin v ms od 0 do 1000

Tabulka 9: Seznam doporučených příkazů

7.6 Podporované funkce

V záložce *Node* se nachází čtyři tlačítka: Node Number, Echo, Send CAN Frame a Read All, které mají zajistit informace o aktuálně připojeném modulu.

Adresa modulu – po kliknutí na tlačítko Node Number se smaže popis se jménem modulu, Stavový diagram SD = 2. Odešle se příkaz, viz tabulka 9. Zapne se časovač s periodou 250 ms. Pokud do 250 ms nepřijde správný příkaz z modulu, tak se vypíše chybové hlášení, v druhém případě se z přijaté zprávy získá hodnota adresy, která se zobrazí na popisku.

Echo – slouží ke zjištění, zda modul komunikuje s počítačem. Princip komunikace je stejný jako u adresy modulu jen s tím rozdílem, že se nastaví stavový diagram na SD = 1 a přijímá a odesílá se jiný příkaz podle tabulky tabulka 9.

Odeslání rámce – slouží k odeslání CAN zprávy o datové délce 2 bajty s hodnotou 0xAA. Stavový diagram je roven 3.

Po stisknutí tlačítka **Read All** se automaticky vyčtou všechny důležité informace o modulu na jedno kliknutí, zjistí se adresa, vyčte se počet chyb, nastavení časování a nakonec se pro skenuje celá síť.

V záložce **LED Control** jsou vyznačeny čtyři stavové LED diody, které můžeme ovládat kliknutím na ovládací prvek. Před ovládáním je nutné mít proskenovanou síť a je třeba vybrat modul, který chceme ovládat. Zaškrtávacím políčkem *Set All Nodes* můžeme ovládat celou síť. Pokud by bylo třeba ovládat jen moduly s CM cívkou, nebo bez cívky, může se použít přepínačů v pravé dolní části obrazovky: *Control Nodes with CM Chokes* nebo *Control Nodes without CM Chokes*. Po každé změně LED ovládacího prvku se provede procedura `Led_Change()`, která zjistí stav všech ostatních ovládacích prvků, zjistí se, které moduly se mají ovládat, a pošle se příslušný příkaz pro změnu módu. Nakonec se odešle samotný příkaz pro ovládání LED diod. Spodní čtyři bity v commandu udávají hodnotu LED diod, která je požadovaná. Pro rychlejší rozsvícení, nebo zhasnutí všech LED diod je k dispozici tlačítko *All LEDs ON* nebo *All LEDs OFF*. Pokud se nevybere modul, který se má ovládat, program nás upozorní chybovou hláškou.

V záložce **Send Data** můžeme odeslat z hlavního modulu datovou zprávu o velikosti 0 až 8 datových bajtů. Hodnotu zprávy lze libovolně měnit, a také lze přepínat mezi zobrazením v desítkové, hexadecimální soustavě nebo zobrazením jako znak ascii. Samotný rámec pro odeslání datové zprávy se skládá z hlavičky ‚S‘ a následuje command, který obsahuje počet datových bajtů a pak následuje přesně definovaný počet odesílaných datových bajtů.

Záložka **Network** umožňuje pomocí tlačítka *Scan Network* proskenovat síť a zjistit, jaké adresy mají právě připojené moduly. Po stisku tlačítka se nejprve smaže formátování a popisky z komponenty *NodeView* třídy *DataGridView*. Stavový diagram se nastaví na číslo 4, zapne se časovač s periodou 1000 ms, po uplynutí této doby

časovač se vypne a přejde se do stavu idle. Při přijmutí a ověření správnosti dat z mikroprocesoru se do komponenty *NodeView* barevně zakreslí adresy modulů, jenž jsou připojeny do sítě. Zeleně je zakreslený modul bez cívky a modře modul s cívkou. Po uplynutí jedné vteřiny od zapnutí časovače se ukončí příjem dat, spočítá se celkový počet modulů, který se vypíše do stavového popisu a přejde se do nečinného stavu, ve kterém se čeká na další příkaz od uživatele.

V záložce **Testing** jsou dvě tlačítka *Test* a *Continual*, pomocí kterých můžeme spouštět testy sítě. Před testem je nutné provést skenování sítě. Tlačítko *Test* spustí pouze jeden test, kdežto tlačítko *Continual* je pro několikanásobné testování sítě v závislosti na hodnotě posuvníku *TBTests*. Dva přepínače slouží ke zvolení testování s arbitrací nebo bez arbitrace. Počet testů se může nastavit od 1 do 50 000. Při stisku tlačítka *Continual* se stavový diagram nastaví na hodnotu 6 a do pomocných proměnných se uloží horní a spodní bajt z hodnoty posuvníku. Odešle se příkaz začínající hlavičkou 'I' hodnota commandu určuje, zda se bude testovat s arbitrací nebo bez a následující dva bajty určují počet testů. Po odeslání tohoto příkazu se vypočítá z rychlosti komunikace a počtu modulů předpokládaná doba ukončení testování. Zapne se časovač, který odpočítá dobu do konce testování. V průběhu testování může program obdržet příkaz značící chybu testování. Vypne se časovač, vypíše se chybové hlášení a program se uvede do výchozího stavu. V případě, že testování proběhlo úspěšně, je tento stav zobrazen v programu.

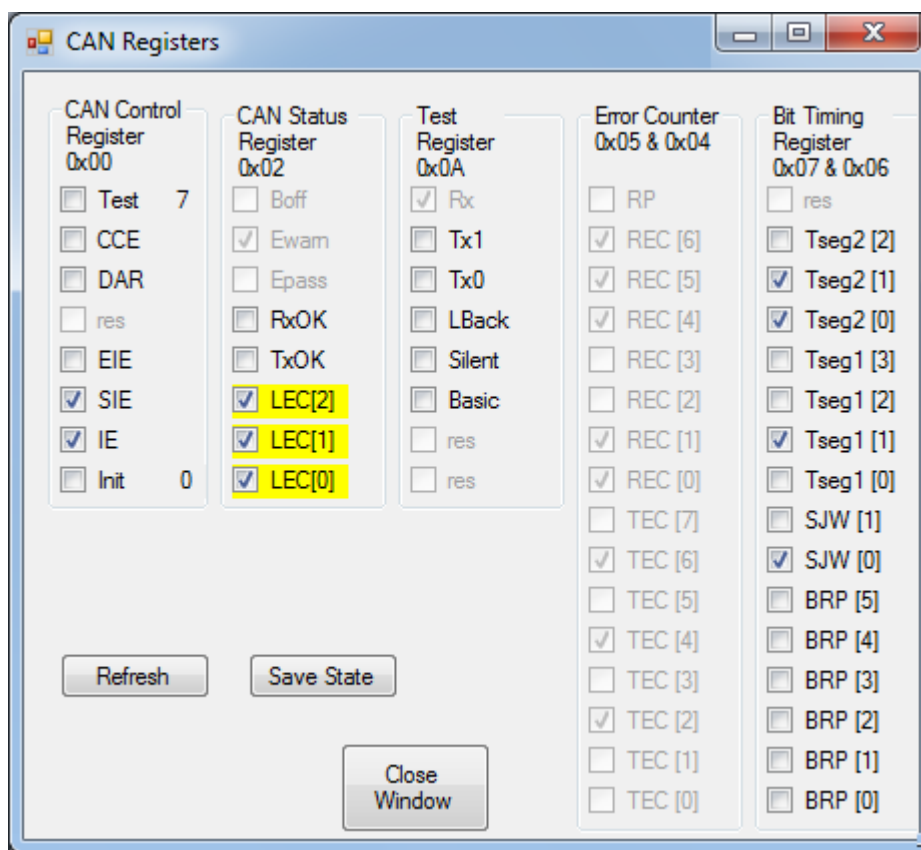
Případné chyby způsobené testováním nebo komunikací po CAN sběrnici lze vyčíst v záložce s názvem **Errors**. Program umožňuje vyčíst počítadlo chyb přímo z CAN řadiče. Tlačítkem *Read Error Counter* se vyčte Status registr, který obsahuje LEC proměnnou. Tato proměnná ukazuje na poslední proběhlou chybu na sběrnici. Error Counter Registr ukazuje aktuální stav chybového čítače. Tlačítkem *Read All Errors* lze vyčíst historii chyb z modulu. Program odešle příkaz podle tabulky 9 a následně se čeká na příjem dat. Formát dat je v pořadí: LEC, REC, TEC, počítadloH, počítadloL, ChybovýModul. Po dokončení odesílání se spočítá celkový počet chyb. Je možné tyto chyby uložit do textového souboru. Tlačítkem *Clear Errors* lze vymazat vnitřní paměť mikroprocesoru kde jsou uloženy vzniklé chyby.

V záložce **Bit Timing** je uvedeno několik důležitých hodnot vedoucích k nastavení rychlosti časování CAN sběrnice. Program dovoluje vyčíst aktuální nastavení pomocí tlačítka *Read* nebo můžeme změnit samostatné časové segmenty či děličku pro CAN časování. V pravé části obrazovky jsou vypsány časové hodnoty jednotlivých segmentů, kdežto v levé části je vidět grafické znázornění Bit Time včetně polohy samplovacího bodu. Program umožňuje změnit pozici samplovacího bodu tažením za popisek či změnit velikost časového segmentu kliknutím na příslušný segment. Segmenty představují přímo vnitřní registry v mikroprocesoru C8051F044, a proto propagační a fázový segment č. 1 je spojen do jednoho segmentu s názvem Seg1. Pro změnu rychlosti CAN sběrnice do aktuálně připojeného modulu slouží tlačítko *Set Master*

Node. Tímto tlačítkem se změní časování pouze pro jeden modul. Pokud chceme změnit nastavení časování pro zbylé moduly, pro tento účel slouží tlačítko *Set Network*.

Záložka **RoundRobin** slouží k vizuálnímu testování modulů. Tlačítko spouští a zastavuje testování, které spočívá v blikání modulů podle zvolené předvolby přepínačů. Rychlost blikání se dá nastavit na posuvníku vpravo. Princip tohoto testování spočívá v periodickém zasílání příkazů příslušným modulům s požadavkem na změnu stavu LED diod. Zpoždění posílání příkazů lze nastavit od 10 ms do 1 s.

V poslední záložce **Settings CCR** lze vyvolat okno, ve kterém jsou zobrazeny stavové registry CAN řadiče. Stav těchto registrů lze vyčíst tlačítkem Refresh. Změna stavu jednotlivých bitů v registru je možná kliknutím na jméno určitého bitu. Pro zápis do CAN řadiče slouží tlačítko *Save State*. Změna stavu je možná jen pro bity, které mají povolené jak čtení, tak i zápis. Registr Error Counter je pouze pro čtení, a tudíž změna stavu není povolena. Pokud změníme stav registrů, je to indikováno žlutým zvýrazněním.



Obrázek 26: Okno programu s CAN registry

7.7 Nastavení rychlosti komunikace

Podle postupu v kapitole 2.5 jsme vypočítali ideální konfiguraci časových registrů pro rychlosti 1 Mbit/s a 500 kbit/s. Délka sběrnice nejvzdálenějších modulů od sebe je 8.6 m. Podle katalogového listu budiče NCV7342 je velikost propagačního zpoždění z TxD pinu na CAN sběrnici rovna 60 ns a propagační zpoždění při příjmu signálu ze sběrnice je rovna 70 ns. Propagační zpoždění na sběrnici se při rychlosti šíření elektrického signálu 6 ns/m rovná 52 ns. Následně můžeme spočítat celkovou velikost propagačního zpoždění, které se rovná

$$t_{\text{PROP_SEG}} = 2(t_{\text{BUS}} + t_{\text{TX}} + t_{\text{RX}}) \text{ [ns]} \quad (12)$$

$$t_{\text{PROP_SEG}} = 2(52 + 60 + 70) \text{ ns}$$

$$t_{\text{PROP_SEG}} = 364 \text{ ns}$$

Rychlost CAN časování je rovna 8 MHz, to znamená, že jedno časové kvantum t_Q se rovná 125 ns bez použití děličky. Vydělením velikosti propagačního zpoždění a časového kvanta určíme velikost propagačního segmentu, který se rovná 3 t_Q . Na oba dva fázové segmenty nám zbyla celkem 4 časová kvanta, jež se ekvivalentně rozdělí mezi oba segmenty. SJW segment nastavíme na velikost 2, stejně jako fázový segment 1.

Přesnost oscilátoru se vypočítá podle vztahu 13 a 14.

$$\Delta f < \frac{t_{\text{SJW}}}{10 \times NBT} = \frac{2}{20 \times 8} = 0.0125 \quad (13)$$

$$\Delta f < \frac{\text{MIN}(\text{fáz. seg1, fáz. seg2})}{2(13 \times NBT - \text{fáz. seg2})} = \frac{2}{2(13 \times 8 - 2)} = 0.0098 \quad (14)$$

Vypočtená požadovaná přesnost oscilátoru je 0,98 %.

Zhodnocení pro rychlost komunikace 1 Mbit/s:

Rychlost komunikace	=	1 Mbit/s
Dělička	=	0
Velikost časového kvanta T_Q	=	125 ns
Počet časových kvant	=	8
Velikost bit time	=	1000 ns
Propagační segment	=	3 T_Q
Fázový segment 1	=	2 T_Q
Fázový segment 2	=	2 T_Q
SJW	=	2 T_Q
Tolerance oscilátoru	=	0.98 %
Pozice samplovacího bodu	=	75 %

Pro rychlost komunikace 500 kbit/s změníme velikost děličky na hodnotu 1. Velikost časového kvanta se zvýší na 250 ns. Po aplikování rovnic z kapitoly 2.5 se hodnoty velikostí propagačních a fázových segmentů neliší od předchozího příkladu s rychlostí 1 Mbit/s, a tudíž pozice samplovacího bodu zůstává na 75 %.

Zhodnocení pro rychlost komunikace 500 kbit/s:

Rychlost komunikace	=	500 kbit/s
Dělička	=	1
Velikost časového kvanta T_q	=	250 ns
Počet časových kvant	=	8
Velikost bit time	=	2000 ns
Propagační segment	=	3 T_q
Fázový segment 1	=	2 T_q
Fázový segment 2	=	2 T_q
SJW	=	2 T_q
Tolerance oscilátoru	=	0.98 %
Pozice samplovacího bodu	=	75 %

8 TESTOVÁNÍ SÍTĚ

Testování jsme prováděli na dvou družích vzorků. První budič byl typu NCV7342-0 a druhý vzorek byl nový, ještě nezveřejněný CAN budič s interním označením 0RDC5-901. Budiče jsme testovali v konfiguraci s použitím cívek souhlasného napětí i bez nich, a nakonec jsme testovali celou síť se všemi připojenými moduly.

8.1 Testování komunikace

Komunikaci jsme testovali v závislosti na různých rychlostech, konfiguracích sítě a samplovacího bodu. Testovali jsme rychlosti 1 Mbit/s, 500 kbit/s, 250 kbit/s a 100 kbit/s. Na těchto rychlostech jsme provedli 50000 cyklů s arbitrací a bez arbitrace a sledovali jsme, zda komunikace proběhla bez chyb nebo s chybami. V každé tabulce je zobrazeno, kolik modulů jsme použili ke komunikaci. Výpis chyb je uveden v příloze. Pod každou tabulkou je uveden rozbor chyb.

8.1.1 Měření číslo 1

Konfigurace: 5 modulů s cívkou souhlasného napětí a budičem typu NCV7342-0. Použité moduly: 0, 1, 2, 3 a 12. Terminace je připojena k modulu 0 a 12. CAN síť řídil modul číslo 0. Celková impedance sběrnice činila 60.3 Ω .

NCV7342-0	1 Mbit/s	500 kbit/s	250 kbit/s	100 kbit/s
arbitrace sb. 75 %	OK	OK	OK	OK
bez arbitrace sb. 75 %	OK	OK	OK	OK
arbitrace sb. 87 %	OK	OK	OK	OK
bez arbitrace sb. 87 %	OK	OK	OK	OK
arbitrace sb. 62 %	OK	OK	OK	OK
bez arbitrace sb. 62 %	OK	OK	OK	OK
arbitrace sb. 50 %	OK	OK	OK	OK
bez arbitrace sb. 50 %	OK	OK	OK	OK
arbitrace sb. 37 %	Chyba	OK	OK	OK
bez arbitrace sb. 37 %	Chyba	OK	OK	OK

Tabulka 10: Hodnoty pro měření číslo 1

Při rychlosti 1 Mbit a nastavení samplovacího bodu na 37% přestala síť fungovat, protože se moduly při pokusu o odpověď dostaly do stavu Bus Off již při skenování sítě. Při nastavení samplovacího bodu na více nebo rovno 50 % testování sítě proběhlo v pořádku.

8.1.2 Měření číslo 2

Konfigurace: 5 modulů bez cívky souhlasného napětí. Moduly byly připojeny pomocí stejných kabelů jako v předchozím případě, abychom mohli porovnat jednotlivá měření. Použité moduly: 4, 5, 6, 7 a 9. Použitý budič byl NCV7342-0. Terminace je připojena na moduly 4 a 9. Síť řídil modul číslo 4. Celková impedance sběrnice činila 60.5 Ω .

NCV7342-0	1 Mbit/s	500 kbit/s	250 kbit/s	100 kbit/s
arbitrace sb. 87 %	OK	OK	OK	OK
bez arbitrace sb. 87 %	OK	OK	OK	OK
arbitrace sb. 75 %	OK	OK	OK	OK
bez arbitrace sb. 75 %	OK	OK	OK	OK
arbitrace sb. 62 %	OK	OK	OK	OK
bez arbitrace sb. 62 %	OK	OK	OK	OK
arbitrace sb. 50 %	OK	OK	OK	OK
bez arbitrace sb. 50 %	OK	OK	OK	OK
arbitrace sb. 37 %	Chyba	OK	OK	OK
bez arbitrace sb. 37 %	Chyba	OK	OK	OK

Tabulka 11: Hodnoty pro měření číslo 2

Testování probíhalo v pořádku až do rychlosti 1 Mbit. Při nastavení samplovacího bodu na 37 % jsme dokázali zjistit stav sítě, ale při testování se komunikace přerušila. Moduly číslo 9 a 4 se pokoušely odesílat CAN zprávu, ale detekovaly se chyby LEC s číslem 3 a 4, což znamená, že na odeslanou zprávu nezareagoval žádný modul a během odesílání zprávy měla být odeslána recesivní úroveň, ale detekována byla dominantní úroveň. Ostatní moduly detekovaly chybu Form Error, kdy přijatý rámeček nebyl ve správném formátu.

8.1.3 Měření číslo 3

Konfigurace: Celkem bylo použito 14 modulů. Z toho 5 modulů má cívku a 9 modulů je bez cívky. Celková impedance sítě činila 59.6 Ω . Pro všechny moduly byl použit budič NCV7342-0.

NCV7342-0	1 Mbit/s	500 kbit/s	250 kbit/s	100 kbit/s
arbitrace sb. 87 %	OK	OK	OK	OK
bez arbitrace sb. 87 %	OK	OK	OK	OK
arbitrace sb. 75 %	OK	OK	OK	OK
bez arbitrace sb. 75 %	OK	OK	OK	OK
arbitrace sb. 62 %	OK*	OK	OK	OK
bez arbitrace sb. 62 %	OK	OK	OK	OK
arbitrace sb. 50 %	Chyba	OK	OK	OK
bez arbitrace sb. 50 %	Chyba	OK	OK	OK

Tabulka 12: Hodnoty pro měření číslo 3

Testování probíhalo v pořádku až do rychlosti 1 Mbit. Při nastavení samplovacího bodu na 62 % a při povolené arbitraci měření proběhlo v pořádku, ale na sběrnici se detekovalo několik chyb. Tyto chyby byly způsobeny nesprávným odesláním rámce z modulu číslo 12. Tento modul detekoval chybu s číslem 4. Podle kapitoly 3.3 se jedná o chybu, kdy modul chtěl odeslat logickou jedničku, ale detekoval logickou nulu. Tento modul vygeneroval aktivní příznak chyby, 6 dominantních bitů, kterými zajistil, že tuto zprávu budou všechny ostatní moduly ignorovat. Protože byla povolená arbitrace, tak modul s číslem 12 odeslal další zprávu, která už byla přenesena bez problému.

Při stejném nastavení samplovacího bodu, ale bez použití arbitrace, nedošlo k žádným chybám. Pravděpodobně to má za následek to, že moduly vysílají pouze ve vymezený čas a nedochází k superpozicím signálu, a tudíž k možnému zkreslení. Při nastavení samplovacího bodu na 50 % dochází k chybám testování.

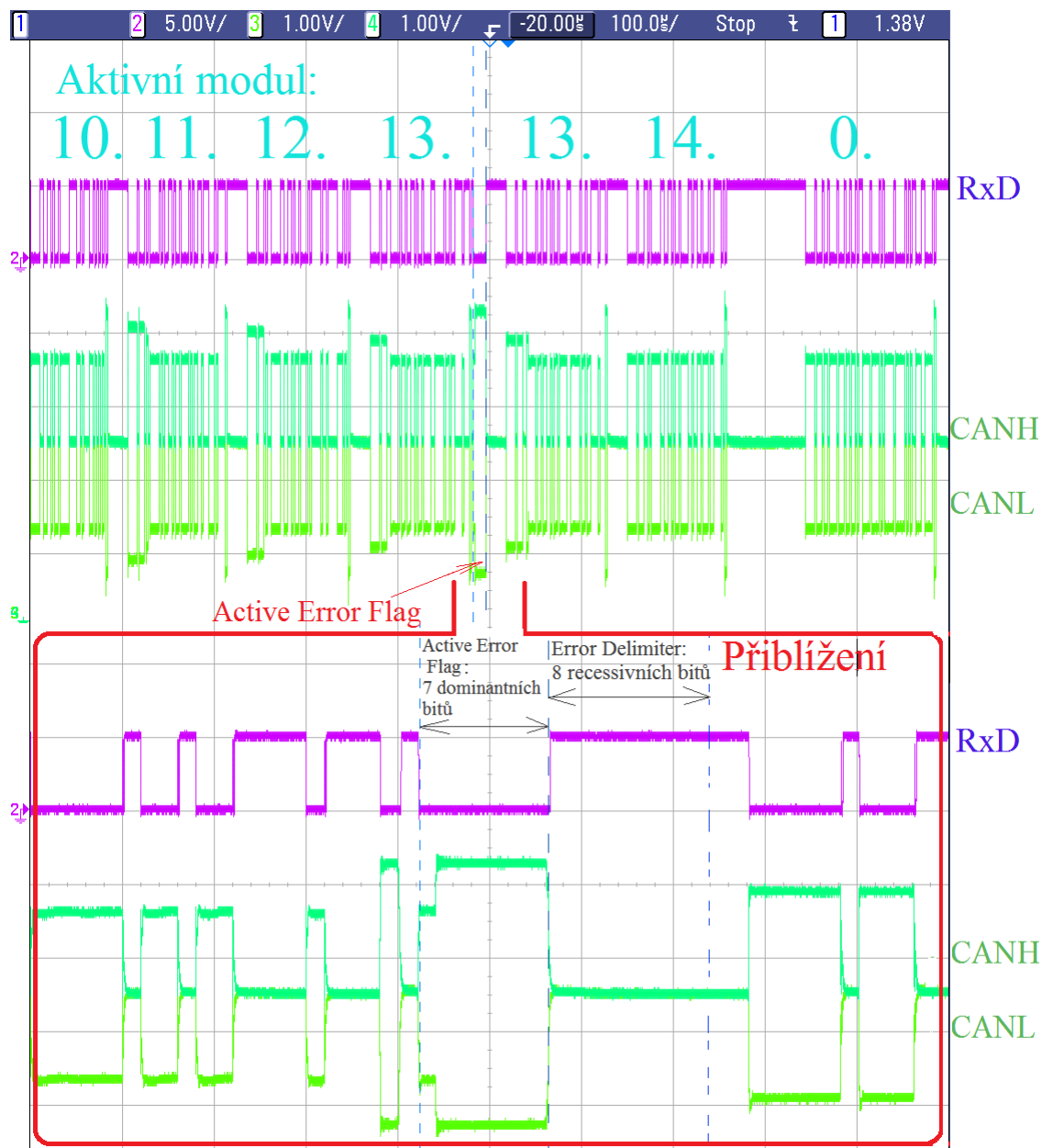
8.1.4 Měření číslo 4

Konfigurace: Celkem bylo použito 14 modulů. Z toho 5 modulů je s cívkou a 9 modulů bez cívky. Celková impedance sítě činila 59.6 Ω . Pro všechny moduly je použit prototyp budiče s interním označením 0RDC5-901.

0RDC5-901	1 Mbit/s	500 kbit/s	250 kbit/s	100 kbit/s
arbitrace sb. 87 %	OK	OK	OK	OK
bez arbitrace sb. 87 %	OK	OK	OK	OK
arbitrace sb. 75 %	OK	OK	OK	OK
bez arbitrace sb. 75 %	OK	OK	OK	OK
arbitrace sb. 62 %	OK	OK	OK	OK
bez arbitrace sb. 62 %	OK	OK	OK	OK
arbitrace sb. 50 %	Chyba	OK	OK	OK
bez arbitrace sb. 50 %	Chyba	OK	OK	OK
arbitrace sb. 37 %	-	OK*	OK	OK
bez arbitrace sb. 37 %	-	OK	OK	OK

Tabulka 13: Tabulka pro měření číslo 4

Při rychlosti 500 kbit/s a nastavení samplovacího bodu na 37 % při arbitraci vznikaly občasné chyby, které ale neměly na testování vliv. Testování na této rychlosti proběhlo v pořádku. Detail průběhu signálů na sběrnici lze vidět na obrázku číslo 27. Při rychlostech 1 Mbit/s a nastavení samplovacího bodu na 50 % se hlavní modul dostal do stavu Bus-Off a komunikace se zastavila. Ostatní moduly detekovaly pouze RxD chyby.



Obrázek 27: Active Error Frame 13. Modulu při 500 kbit/s sp. 37 %

8.1.5 Měření číslo 5

Konfigurace: celkem bylo použito 5 modulů s cívkou. Celková impedance sítě činila 59.6 Ω . Pro všechny moduly byl použit prototyp budiče 0RDC5-901. Použité moduly byly 0, 1, 2, 3 a 12. Terminace byla připojena k modulu 0 a 12. CAN síť řídil modul s číslem 0.

0RDC5-901	1 Mbit/s	500 kbit/s	250 kbit/s	100 kbit/s
arbitrace sb. 87 %	OK	OK	OK	OK
bez arbitrace sb. 87 %	OK	OK	OK	OK
arbitrace sb. 75 %	OK	OK	OK	OK
bez arbitrace sb. 75 %	OK	OK	OK	OK
arbitrace sb. 62 %	OK	OK	OK	OK
bez arbitrace sb. 62 %	OK	OK	OK	OK
arbitrace sb. 50 %	Chyba	OK	OK	OK
bez arbitrace sb. 50 %	Chyba	OK	OK	OK
arbitrace sb. 37 %	Chyba	OK	OK	OK
bez arbitrace sb. 37 %	Chyba	OK	OK	OK

Tabulka 14: Tabulka pro měření číslo 5

Testování při rychlostech 500 kbit/s proběhlo v pořádku. Při rychlostech 1 Mbit/s a nastavení samplovacího bodu na 50 % měření skončilo s chybou, ale moduly se nedostaly do stavu Bus Off, tudíž bylo možno pokračovat v komunikaci. Při nastavení samplovacího bodu na 37 % nebylo možné vyčíst stav sítě a hlavní modul přešel do stavu Bus Off.

8.1.6 Měření číslo 6

Konfigurace: celkem bylo použito 5 modulů bez cívky souhlasného napětí. Celková impedance sítě činila 60.2 Ω . Pro všechny moduly byl použit prototyp budiče 0RDC5-901.

0RDC5-901	1 Mbit/s	500 kbit/s	250 kbit/s	100 kbit/s
arbitrace sb. 87 %	OK	OK	OK	OK
bez arbitrace sb. 87 %	OK	OK	OK	OK
arbitrace sb. 75 %	OK	OK	OK	OK
bez arbitrace sb. 75 %	OK	OK	OK	OK
arbitrace sb. 62 %	OK	OK	OK	OK
bez arbitrace sb. 62 %	OK	OK	OK	OK
arbitrace sb. 50 %	Chyba	OK	OK	OK
bez arbitrace sb. 50 %	Chyba	OK	OK	OK
arbitrace sb. 37 %	Chyba	OK	OK	OK
bez arbitrace sb. 37 %	Chyba	OK	OK	OK

Tabulka 15: Tabulka pro měření číslo 6

Výsledek měření byl podobný jako při měření číslo 5. Testování při rychlostech 500 kbit/s proběhlo v pořádku. Při rychlostech 1 Mbit/s a nastavení samplovacího bodu na 50 % měření skončilo s chybou, že hlavní modul nedostal všechny požadované odpovědi.

9 ZÁVĚR

Cílem zadání bylo navrhnout a vytvořit obvodové uspořádání testovacích modulů, které simulují typické zařízení v automobilech komunikující pomocí sběrnice CAN pro účely testování funkčnosti nových CAN obvodů.

Podařilo se mi navrhnout desku plošných spojů pro modul s CAN budičem. Těchto modulů bylo vyrobeno, osazeno a oživeno čtrnáct. Po zapojení těchto modulů do sběrnice si uživatel vybere jeden modul, který se propojí s počítačem a pomocí počítačového programu se může tímto modulem ovládat celou síť. V uživatelském programu jsou implementovány funkce jako např. zjištění adres všech připojených modulů, odesílání datových zpráv do sítě, periodické testování komunikace, čtení a vyhodnocení chyb, změna nastavení časování sběrnice a čtení a nastavování konfiguračních registrů.

Pokud není k dispozici počítač, je možné pomocí přepínače nastavit jeden modul do pozice master a automaticky začne probíhat vizuální testování, které spočívá v postupném blikání LED diod, jež řídí přes CAN sběrnici master modul.

Pro otestování funkčnosti jsem si zvolil dva typy budičů NCV7342-0 a prototyp s označením 0RDC5-901. Testování jsem prováděl pro různé rychlosti sběrnice od 100 kbit/s do 1 Mbit/s a pro všechny možné nastavení samplovacího bodu. Z výsledků můžeme vidět, že nová verze čipu 0RDC5-901 pro správnou funkci vyžaduje posunutí samplovacího bodu o 12.5 %. Nicméně pro námi zvolenou konfiguraci sítě je doporučené nastavení samplovacího bodu 75 % celkové doby jednoho bitu. Můžeme tedy říct, že oba dva čipy tuto podmínku splňují a testování proběhlo v pořádku.

Literatura

- [1] DAVES, Robert, Alan BURNS, Reinder BRIL a Johan LUKKIEN. *Controller Area Network (CAN) schedulability analysis: Refuted, revisited and revised*. 2007, s. 34. Dostupné z: <http://goo.gl/LNrZPF>
- [2] BOSCH: *C_CAN User's Manual*. Automotive Equipment Division 8, Revision 1.2, 45 s.
- [3] ISO: International Organization for Standardization. *ISO 11898-1:2003* [online]. 2009, 2013 [cit. 2014-03-03]. Dostupné z: http://www.iso.org/iso/catalogue_detail.htm?csnumber=33422
- [4] CAN in Automation (CiA). *CAN physical layer* [online]. 2001-2014 [cit. 2014-03-03]. Dostupné z: <http://www.can-cia.org/index.php?id=systemdesign-can-physicallayer>
- [5] KOCOUREK, P. Sběrníkové systémy letadel. *Controller Area Network (CAN)* [online]. 2008 [cit. 2014-03-03]. Dostupné z: <http://measure.feld.cvut.cz/cs/system/files/files/cs/vyuka/predmety/x38ssl/CANPopis.pdf>
- [6] ZEŽULKA, František, Petr FIEDLER a Zdeněk BRADÁČ. *Prostředky průmyslové automatizace*. Brno, 2002. VUT, Fakulta Elektrotechniky a Komunikačních Technologií.
- [7] BOSCH: *CAN Specification Version 2.0*, Stuttgart, 1991, 72 s.
- [8] CORRIGAN, Steve. TEXAS INSTRUMENTS. SLLA270. *Controller Area Network Physical Layer Requirements*. Dallas, 2008, 15 s. Dostupné z: <http://www.ti.com/lit/an/slla270/slla270.pdf>
- [9] LEPKOWSKI, J a B WOLFE. ON SEMICONDUCTOR. *EMI/ESD protection solutions for the CAN bus*. 2005. CAN in Automation.
- [10] ETSCHBERGER, Konrad. *Controller area network: basics, protocols, chips and applications*. Weingarten: IXXAT Press, 2001, 431 s. ISBN 30-000-7376-0.
- [11] SILICON LABORATORIES. *C8051F040/1/2/3/4/5/6/7* [online]. Austin, 2004, 328 s. [cit. 2014-03-04]. Dostupné z: <http://www.ecse.rpi.edu/courses/CStudio/Silabs/C8051F04x.pdf>
- [12] POLÁK, Josef. AMI SEMICONDUCTOR. *CAN Application Board*. Brno, 2006, 56 s.
- [13] ON SEMICONDUCTOR. *NCV7340 High Speed Low Power CAN Transceiver* [online]. Denver, 2013, 12 s. [cit. 2014-03-27]. Rev. 7 Dostupné z: http://www.onsemi.com/pub_link/Collateral/NCV7340-D.PDF
- [14] DRAKA Cable Wuppertal GmbH. *FL2X11Y 299* [online]. Wuppertal, 2002. [cit. 2014-03-28]. Dostupné z: http://www.prysmiangroup.com/en/business_markets/markets/automotive/downloads/datasheets/DB-F-003F2002-FL2X11Y-299.pdf
- [15] FOX ELECTRONICS. *Quartz Crystal Design Notes* [online]. Fort Myers, 2004. [cit. 2014-03-29]. Dostupné z: <http://www.foxonline.com/pdfs/xtaldesignnotes.pdf>

- [16] AN1798. *CAN Bit Timing Requirements*. Rev.4. East Kilbride: Freescale Semiconductor, 2004. Dostupné z: http://cache.freescale.com/files/microcontrollers/doc/app_note/AN1798.pdf
- [17] COOK, J a J FREUDENBERG. *Controller Area Network (CAN)*. In: THE INSTITUTE FOR DYNAMIC SYSTEMS AND CONTROL. [online]. Zurich, 2008 [cit. 2014-04-23]. Dostupné z: http://www.idsc.ethz.ch/Courses/embedded_control_systems/Exercises/ControlAreaNetwork08.pdf
- [18] VESELÝ, Rudolf. MCONTROLLERS.COM. *CAN Bus*. s. 10. Dostupné z: <http://forum.mcontrollers.com/download.php?id=614&sid=8f9c1d31f7a9a9be644136252e0204be>
- [19] ON SEMICONDUCTOR. *NCV7342 High Speed Low Power CAN Transceiver* [online]. Denver, 2014, 15 s. [cit. 2014-03-27]. Rev. 2 Dostupné z: http://www.onsemi.com/pub_link/Collateral/NCV7342-D.PDF
- [20] HARTWICH, Florian, Bernd MÜLLER, et al. *Timing in the TTCAN Network*. In: Stuttgart. Dostupné z: http://www.bosch-semiconductors.de/media/pdf_1/canliteratur/timing_in_ttcan.pdf

Seznam zkratek

Zkratka	význam
BRP	Baud Rate Prescaler
CAN	Controller Area Network
CCE	Configuration Change Enable
CiA	CAN in Automation
CRC	Cyclic Redundancy Check
DAR	Disable Automatic Retransmission
DLC	Data Length Code
DPS	Deska plošných spojů
ESD	Electrostatic Discharge
EMC	Electromagnetic Compatibility
MCU	Microcontroller
OSBDM	Open Source Background Debug Mode interface
PC	Personal Computer
REC	Receive Error Counter
SJW	Synchronization Jump Width
TDMA	Time division multiple access
TEC	Transmit Error Counter
Tseg1	Časový segment před samplovacím bodem
Tseg2	Časový segment za samplovacím bodem
TTCAN	Time Triggered CAN

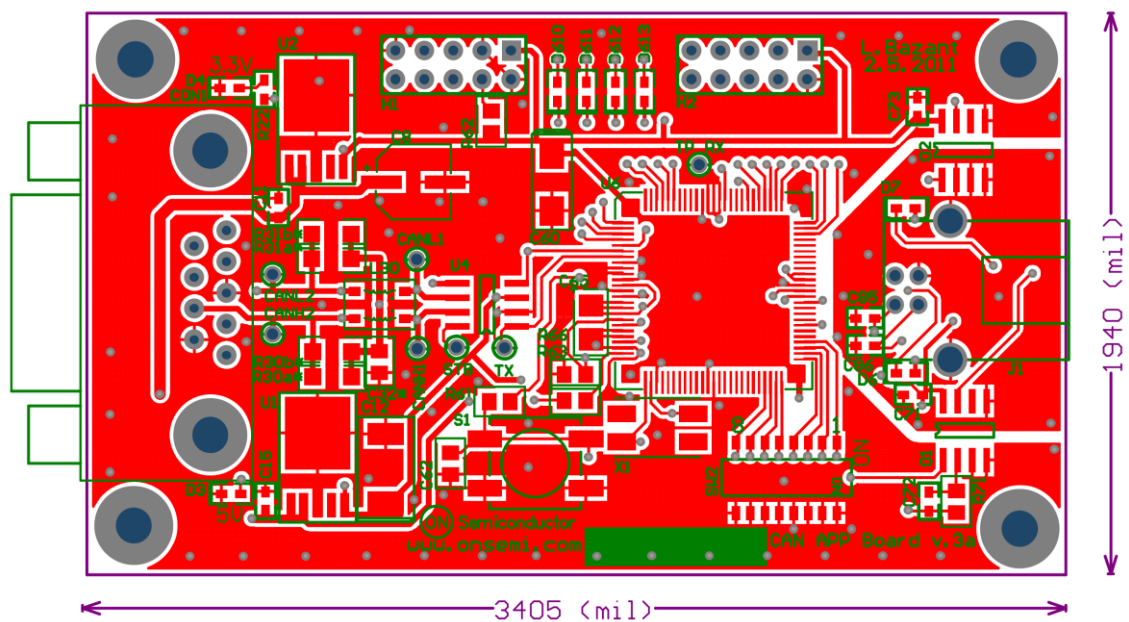
Seznam příloh

Příloha 1. DPS

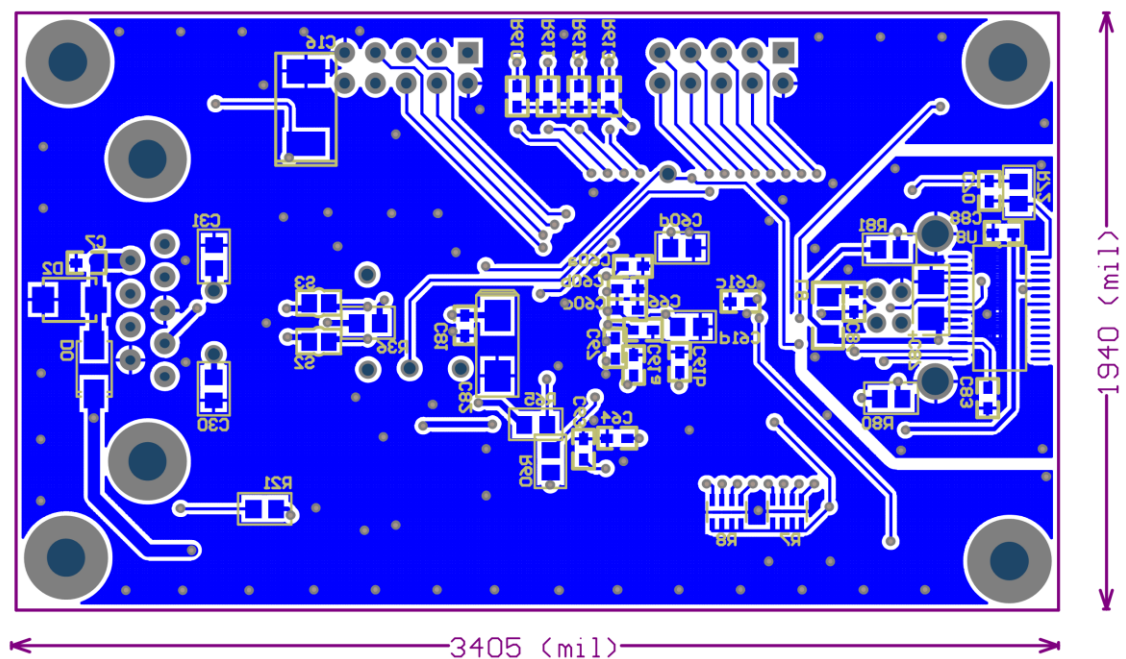
Příloha 2. Výpis hodnot z testování

Příloha 3. CD/DVD

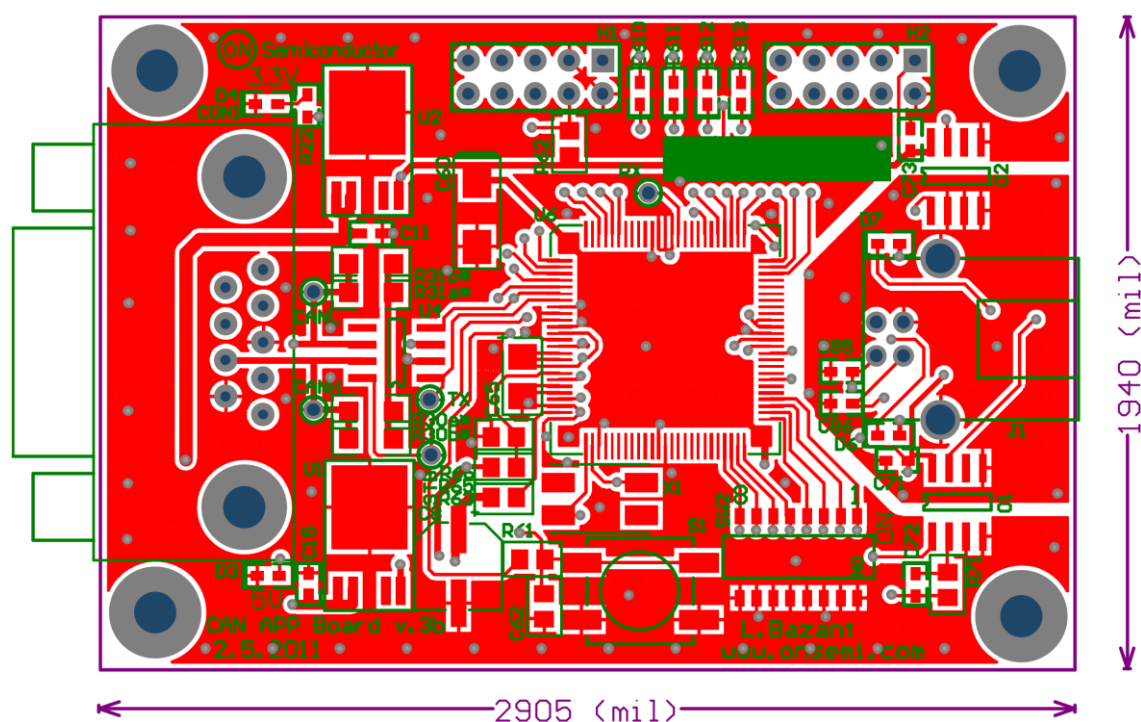
Příloha 1



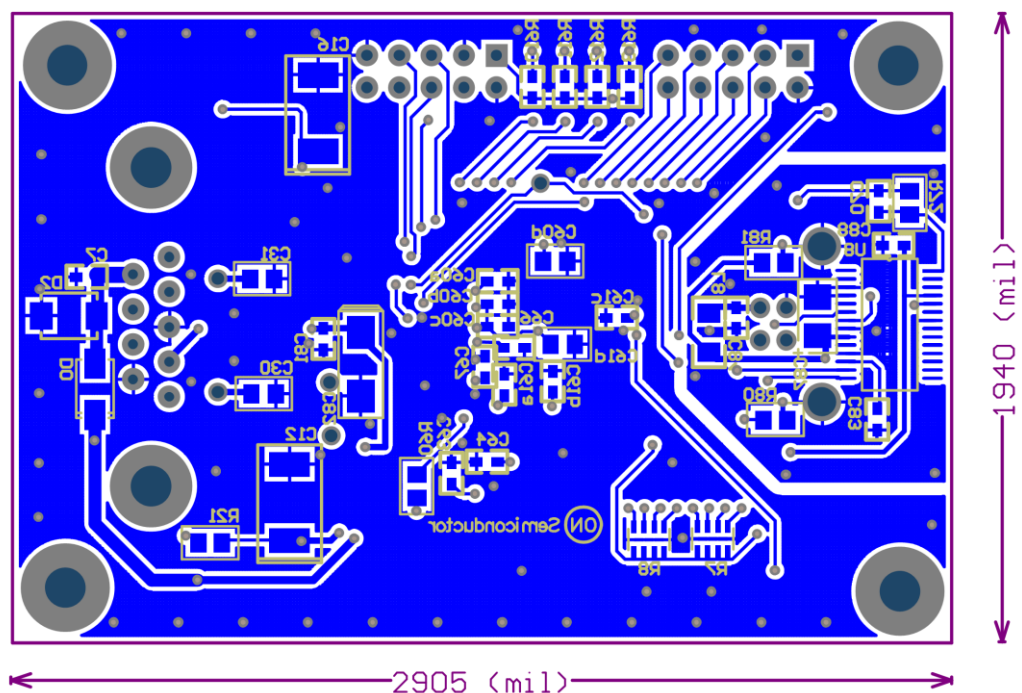
A – Vrchní vrstva desky CAN APP Board V.3a



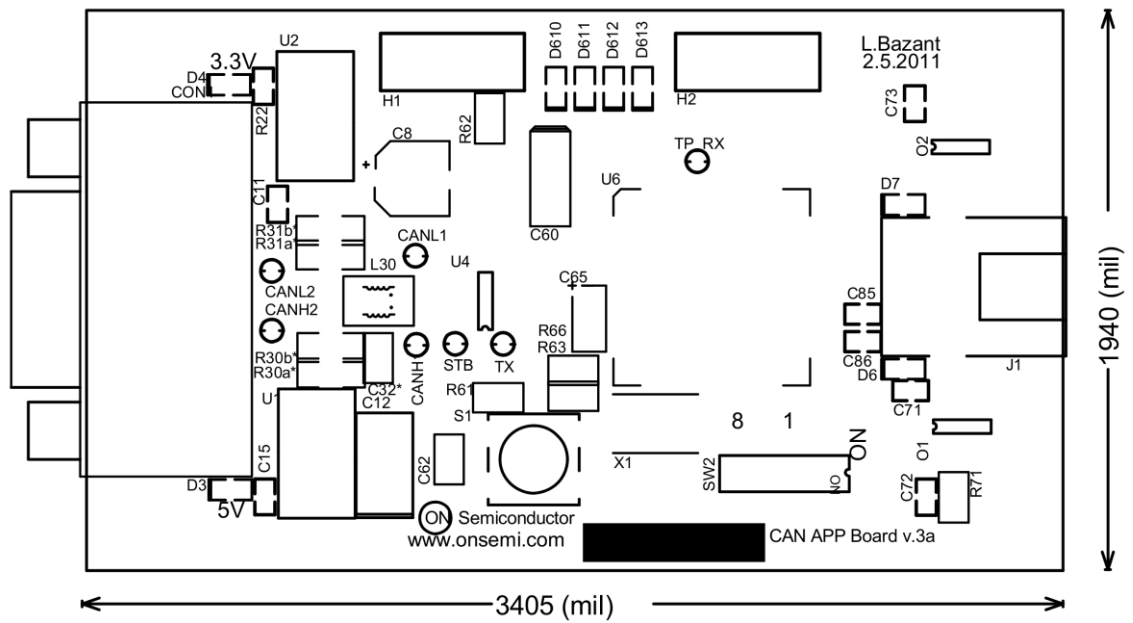
B – Spodní vrstva desky CAN APP Board V.3a



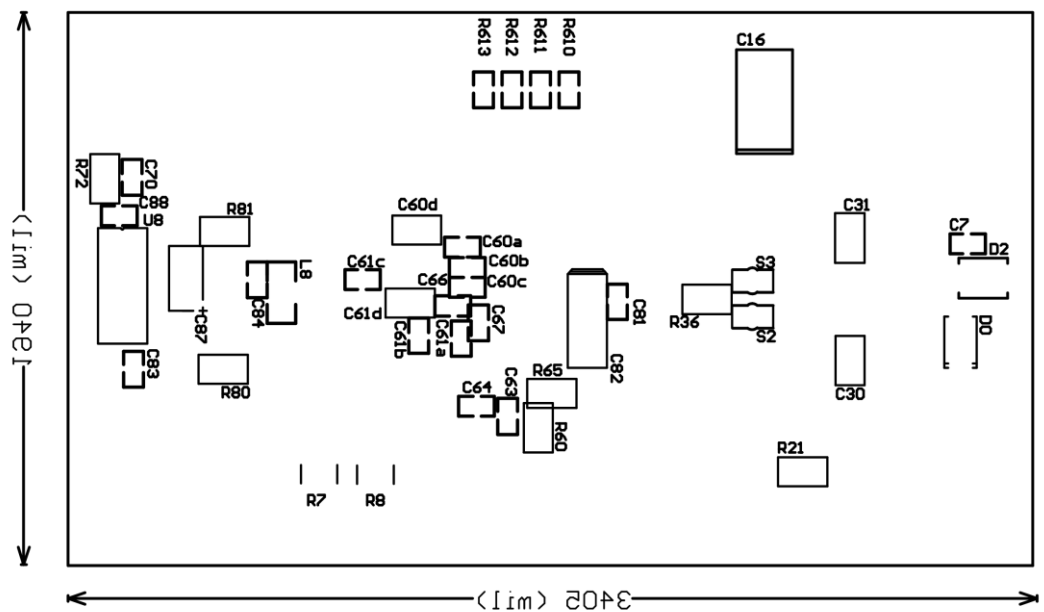
C – Vrchní vrstva desky CAN APP Board V.3b



D – Vrchní vrstva desky CAN APP Board V.3b



E – Osazovací plán vrchní vrstvy CAN APP Board V.3a



F – Osazovací plán spodní vrstvy CAN APP Board V.3a

Příloha 2:

Test. Cyklus	Modul	TEC	REC	LEC
0	2	8	0	4
0	2	16	0	4
0	2	24	0	4
0	2	32	0	4
0	2	40	0	4
0	2	48	0	4
0	2	56	0	4
0	2	64	0	4
0	2	72	0	4
0	2	80	0	4
0	2	88	0	4
0	2	96	0	4
0	2	104	0	4
0	2	112	0	4
0	2	120	0	4
0	2	128	0	4
0	2	136	0	4
0	2	144	0	4
0	2	152	0	4
0	2	160	0	4
0	2	168	0	4
0	2	176	0	4
0	2	184	0	4
0	2	192	0	4
0	2	200	0	4
0	2	208	0	4
0	2	216	0	4
0	2	224	0	4
0	2	232	0	4
0	2	240	0	4
0	2	248	0	4
0	2	248	0	4

Tabulka 1. Měření číslo 1. Rychlost 1 Mbit/s sb. 37 % Modul 0

Test. Cyklus	Modul	TEC	REC	LEC
5	3	0	1	2
7	3	0	1	2
8	3	0	1	2
9	3	0	1	2
10	3	0	1	2
12	3	0	1	2
13	3	0	1	2
14	3	0	1	2
15	3	0	1	2

19	2	8	0	4
19	2	16	0	4
19	2	24	0	4
19	2	32	0	3
19	2	40	0	4
19	2	48	0	3
19	2	56	0	4
19	2	64	0	3
19	2	72	0	4
19	2	80	0	3
19	2	88	0	4
19	2	96	0	4
19	2	104	0	3
19	2	112	0	3
19	2	120	0	4
19	2	128	0	3
19	2	136	0	4
19	2	144	0	3
19	2	152	0	4
19	2	160	0	4
19	2	168	0	3
19	2	176	0	4
19	2	184	0	4
19	2	192	0	3
19	2	200	0	4
19	2	208	0	4
19	2	216	0	3
19	2	224	0	4
19	5	223	1	2
20	2	222	1	1
75	5	167	1	2
96	5	146	0	4
96	5	154	0	4
96	5	162	0	3
96	5	170	0	4
97	2	169	1	3
97	2	177	1	4
97	2	176	1	2
99	4	180	0	4
99	4	196	0	4
99	4	204	0	4
99	4	212	0	4
99	4	228	0	4
99	4	228	1	1
99	4	228	2	1
99	4	228	4	1
99	4	228	5	1
99	4	228	7	1
99	4	228	8	1
99	4	228	10	1

99	4	228	12	1
99	4	228	13	1
99	4	228	15	1
99	4	228	17	2
99	4	228	26	1
99	4	228	27	2
99	4	228	36	1
99	4	228	37	2
99	4	228	46	1
99	4	228	47	2
99	4	228	56	1
99	4	228	57	2
99	4	228	66	1
99	4	228	67	2
99	4	228	77	1
99	4	228	77	2
99	4	228	86	1
99	4	228	87	2
99	4	228	97	1
99	4	228	97	2
99	4	228	106	1
99	4	228	107	2
99	4	228	116	1
99	4	228	117	2
99	4	228	126	1
99	4	228	127	2
99	4	228	127	1
99	4	228	127	2
99	4	244	127	3
99	4	252	127	4
99	4	252	127	4
99	4	252	127	4

Tabulka 2. Měření číslo 2. Rychlost 1 Mbit/s sb. 37 % Modul 4

Test. Cyklus	Modul	TEC	REC	LEC
67	2	0	1	1
2845	2	0	1	1
7114	2	0	1	1
9431	2	0	1	1
12429	2	0	1	1
13605	2	0	1	1
17827	2	0	1	1
21061	13	0	1	1
22841	13	0	1	1
25255	2	0	1	1
29719	13	0	1	1

29869	2	0	1	1
31517	2	0	1	1
32610	2	0	1	1
36215	2	0	1	1

Tabulka 3. Měření číslo 3. Rychlost 1 Mbit/s sb. 62 % Modul 0

Test. Cyklus	Modul	TEC	REC	LEC
2544	2	0	1	1
3747	4	0	1	1
5350	2	0	1	1
7050	2	0	1	1
7363	2	0	1	1
9209	2	0	1	1
10102	13	0	1	1
10171	2	0	1	1
12799	2	0	1	1
13965	2	0	1	1
14157	13	0	1	1
18410	2	0	1	1
18410	2	0	2	1
18410	2	0	3	1
18410	2	0	4	1
18410	2	0	5	1
18410	2	0	6	1
18410	2	0	7	1
18410	2	0	8	1
18410	2	0	9	1
18410	2	0	10	1
18410	2	0	11	1
18410	2	0	12	1
18410	2	0	13	1
18410	2	0	14	1
18410	2	0	15	1
21177	2	0	1	1
28476	2	0	1	1
28510	4	0	1	1
29525	2	0	1	1
31976	2	0	1	2
33856	2	0	1	1
33856	2	0	1	2
35856	13	0	1	1
35856	2	0	2	1
35856	2	0	3	1
35856	2	0	4	1
35856	2	0	5	1
35856	2	0	6	1
35856	2	0	7	1

35856	2	0	8	1
35856	2	0	9	1
35856	2	0	10	1
35856	2	0	11	1
35856	2	0	12	1
35856	2	0	13	1
35856	2	0	14	1
35856	2	0	15	1
36834	2	0	1	1
36835	4	0	1	1
37404	2	0	1	1
38587	2	0	1	2
40517	13	0	1	1
41391	2	0	1	1
42236	2	0	1	1

Tabulka 4. Měření číslo 4. Rychlost 500 kbit/s sb. 37 % Modul 1

Test. Cyklus	Modul	TEC	REC	LEC
0	-	32	127	4
0	-	32	127	2
0	-	40	127	4
0	-	40	127	2
0	-	48	127	4
0	-	48	127	1
0	-	48	127	4
0	-	56	127	4
0	-	56	127	2
0	-	64	127	4
0	-	64	127	2
0	-	72	127	4
0	-	72	127	2
0	-	80	127	4
0	-	80	127	1
0	-	80	127	4
0	-	88	127	1
0	-	88	127	2
0	-	96	127	4
0	-	96	127	2
0	-	104	127	4
0	-	104	127	2
0	-	112	127	4
0	-	112	127	2
0	-	120	127	4
0	-	120	127	1
0	-	120	127	4
0	-	128	127	4
0	-	128	127	2
0	-	136	127	4

0	-	136	127	2
0	-	144	127	4
0	-	144	127	2
0	-	152	127	4
0	-	152	127	1
0	-	152	127	4
0	-	160	127	1
0	-	160	127	2
0	-	168	127	4
0	-	168	127	2
0	-	176	127	4
0	-	176	127	2
0	-	184	127	4
0	-	184	127	2
0	-	192	127	4
0	-	192	127	1
0	-	192	127	4
0	-	200	127	4
0	-	200	127	2
0	-	208	127	4
0	-	208	127	2
0	-	216	127	4
0	-	216	127	2
0	-	224	127	4
0	-	224	127	1
0	-	224	127	4
0	-	232	127	1
0	-	232	127	2
0	-	240	127	4
0	-	240	127	2
0	-	248	127	4
0	-	248	127	4

Test. Cyklus	Modul	TEC	REC	LEC
0	2	8	0	3
209	5	0	1	2
550	5	0	1	2
1462	5	0	1	2